

NASA TECHNICAL NOTE



NASA TN D-6793

c.1

LOAN COPY: RETURN
AFWL (DOUL)
KIRTLAND AFB, N.



NASA TN D-6793

**ROOTS OF POLYNOMIALS BY RATIO
OF SUCCESSIVE DERIVATIVES**

by James E. Crouse and Charles W. Putt

Lewis Research Center

Cleveland, Ohio 44135





0133731

1. Report No. NASA TN D-6793		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle ROOTS OF POLYNOMIALS BY RATIO OF SUCCESSIVE DERIVATIVES				5. Report Date May 1972	
				6. Performing Organization Code	
7. Author(s) James E. Crouse and Charles W. Putt				8. Performing Organization Report No. E-6585	
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135				10. Work Unit No. 764-74	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Technical Note	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract <p>An order of magnitude study of the ratios of successive polynomial derivatives yields information about the number of roots at an approached root point and the approximate location of a root point from a nearby point. The location approximation improves as a root is approached, so a powerful convergence procedure becomes available. These principles are developed into a computer program which finds the roots of polynomials with real number coefficients.</p>					
17. Key Words (Suggested by Author(s)) Roots of polynomials Polynomial Root Complex				18. Distribution Statement Unclassified - unlimited	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 81	
				22. Price* \$3.00	

ROOTS OF POLYNOMIALS BY RATIO OF SUCCESSIVE DERIVATIVES

by James E. Crouse and Charles W. Putt

Lewis Research Center

SUMMARY

Computer programs for finding roots of polynomials often give unsatisfactory answers where roots relatively close together are encountered. This difficulty to a large extent can be avoided with a procedure utilizing ratios of successive derivatives. The specific information gained from the ratio of successive derivatives is the number of roots at the root point approached and the approximate location of a trail point with respect to the closest root. The location approximation improves as a root is approached so a powerful convergence procedure becomes available.

Equations are developed in this report for the general case of polynomials with complex number coefficients. Stepwise procedures are given for obtaining accurate roots for the general case. These principles are developed into a computer program which finds the real and complex number roots of polynomials for the special case of real number coefficients. Some examples are shown to illustrate the root resolution capability of the program.

INTRODUCTION

High-speed computing has made some formerly laborious mathematical procedures, such as solving for the roots of rather high degree polynomials, somewhat more practical for broad engineering application. Before the prominence of computers the solution of high degree polynomials for roots had an element of art to complement the science. The solution of general polynomials on computers, however, requires completely logical steps. Many methods have been developed and programmed for general use (e.g., refs. 1 and 2). Almost all of these programs use iterative procedures and require the evaluation of the polynomial at each trial root. Most of the programs work well for the vast majority of cases; however, they usually either compute an inaccurate solution or fail to converge to a solution for some root combinations. These difficulties are usually caused by multiple roots at a point or by two or more very close roots.

Since roots are defined as the points for which a polynomial equals zero, iterative root finding techniques search for points that give a polynomial value of zero. When roots are not close together a polynomial will have significant slope at a root; so a tolerance of the closeness of the polynomial to zero can be and effectively is used as a root criterion. However, when roots are very close together or when multiple roots occur at a point, the polynomial approaches these roots at very nearly zero slope. With these low slopes, very poor root resolution capability is possible with a polynomial-equal-zero tolerance criterion. And in some cases, programs fail to converge at all. In most cases of failure, either the polynomial cannot be evaluated near a root with sufficient accuracy or the polynomial value over large domains of the complex plane is outside the range of numbers representable on a computer. Thus, it appears that knowledge about the relative closeness of the approached root to other roots is needed for more comprehensive root finding computer programs.

Polynomial derivatives give a clue as to the nature of the root or roots approached. In fact, the ratios of successive polynomial derivatives give the following very useful information: (1) the multiplicity of a root; that is, the number of roots at the root point approached, (2) the closeness of a trial point to the root approached, and (3) a good approximation as to where the next nearest root is when at a root. Thus, an approach using the ratios of successive polynomial derivatives offers the possibility of accurate roots-of-polynomial computer programs with very high reliability.

In this report the general principles of what can be learned from the ratios of polynomial derivatives (including polynomials with complex number constants) is presented and discussed. Then these principles are used in a computer program which finds the roots of polynomials for the special case of real number coefficients. The program is included in the report along with examples of input, output, and resolution capabilities.

DEVELOPMENT OF THE GENERAL METHOD

General Procedure

In the following development of equations, the polynomial is assumed to be of the general form. (Symbols are defined in appendix A.)

$$P(z) = a_0 + a_1z + a_2z^2 + a_3z^3 + \dots + a_jz^j + \dots + a_nz^n \quad (1)$$

where the variable z and the constants a_j for $j = 0$ to n are complex numbers in the general case. The polynomial in terms of its roots can be written as

$$P(z) = (z - b_1)(z - b_2)(z - b_3) \dots (z - b_j) \dots (z - b_n) \quad (2)$$

Equations (1) and (2) are always analytic. Thus, a derivative always exists, and it has the same value at a point independent of the direction of approach.

The first derivative of equation (2) is

$$P'(z) = \left\{ [(z - b_2)(z - b_3) \dots (z - b_n)] + [(z - b_1)(z - b_3) \dots (z - b_n)] \right. \\ \left. + \dots + [(z - b_1)(z - b_2) \dots (z - b_{n-1})] \right\} \quad (3)$$

The first derivative ratio is formed by dividing equation (3) by equation (2); the result is

$$\frac{P'(z)}{P(z)} = f_1 = \frac{1}{z - b_1} + \frac{1}{z - b_2} + \frac{1}{z - b_3} + \dots + \frac{1}{z - b_n} \quad (4)$$

Equation (4) has n terms, but note that as a root is approached a few or usually only one term will predominate. If only one term predominates, equation (4) can be approximated by

$$\frac{P'(z)}{P(z)} = f_1 \approx \frac{1}{z - b_j} \quad (5)$$

at a trial z in the vicinity of b_j . Equation (5) then can be solved for b_j to give a much closer z approximation to the root in the next iteration. As z gets closer to b_j , the predominance of the one term in equation (4) becomes increasingly outstanding. Thus, it is possible to close in on the root point very rapidly.

The value of knowing $z - b_j$ near a root has been indicated; but so far, only the special case of a point near a single root has been covered. For the general case of arbitrary z in $P(z)$, z may not be relatively close to any one root point so no one term in equation (4) predominates. Another less common possibility is the nearest root point may have multiple roots m , so that m terms in equation (4) sum to $m/(z - b_j)$. Thus, for general analysis, f_1 can be expressed as

$$\frac{P'(z)}{P(z)} = f_1 = \frac{m}{z - b_j} + g_1 \quad (6)$$

where b_j is the closest root, m is number of roots at b_j , and g_1 is the sum of the rest of the terms, $(n - m)$, in equation (4). Information about m and g_1 is needed before a value of b_j can be calculated from equation (6).

More information about m and g_1 can be obtained from equations involving higher order derivative ratios. The development of these equations, however, becomes more

complicated as the order of the derivative increases; so the stepwise derivations through the fourth derivative are shown in appendix B. Particular equations developed there will be drawn into the text as needed.

The second derivative ratio (eq. (B10)) is

$$\frac{P''(z)}{P'(z)} = \frac{\frac{m(m-1)}{(z-b_j)^2} + \frac{2mg_1}{z-b_j} + g_1^2 - g_2}{\frac{m}{z-b_j} + g_1} \quad (7)$$

This is another equation in m , g_1 , and a new variable g_2 , which is a summation of terms of the form $1/(z-b_j)^2$ for all b_j values exclusive of the nearest root points. The introduction of the new variable indicates that more equations of this type will not give a set that can reasonably be solved in a direct mathematical way. Much implicit information, however, can be obtained from an order of magnitude study of the terms in the derivative ratios.

Order of Magnitude Studies of Derivative Ratios

Let us begin by observing what happens to the first and second derivative ratios (eqs. (6) and (7)) as a single root ($m = 1$) is approached. First note that as $z \rightarrow b_j$ in equation (6), $P'(z)/P(z) \rightarrow \pm\infty$.

Now consider equation (7). The first term in the numerator is zero since $m = 1$. As $z \rightarrow b_j$, equation (7) essentially reduces to

$$\frac{P''(z)}{P'(z)} \approx \frac{\frac{2mg_1}{z-b_j}}{\frac{m}{z-b_j}} = 2g_1 \quad (8)$$

Thus, as $z \rightarrow b_j$ for $m = 1$, the second derivative ratio approaches $2g_1$, which will be virtually a constant for small z changes near b_j . Consequently, for $m = 1$, the first derivative ratio increases in magnitude rapidly and the second derivative ratio approaches $2g_1$ as $z \rightarrow b_j$.

Now let us consider the case of $m > 1$. The first derivative ratio still approaches $\pm\infty$ as $z \rightarrow b_j$. But, in this case equation (7) essentially reduces to

$$\frac{P''(z)}{P'(z)} \approx \frac{\frac{m(m-1)}{(z-b_j)^2}}{\frac{m}{z-b_j}} = \frac{m-1}{z-b_j} \quad (9)$$

Thus the second derivative ratio also approaches $\pm\infty$ as $z \rightarrow b_j$ for $m > 1$. In fact, the ratio between the magnitude of the second and first derivative ratios is $(m-1)/m$ for $m > 1$.

At this point the third derivative ratio is lifted from the appendix (eq. (B15)) to show the following pattern that is developing:

$$\frac{P'''(z)}{P''(z)} = \frac{\frac{m(m-1)(m-2)}{(z-b_j)^3} + \frac{3m(m-1)g_1}{(z-b_j)^2} + \frac{3m(g_1^2 - g_2)}{z-b_j} + g_1^3 - 3g_1g_2 + 2g_3}{\frac{m(m-1)}{(z-b_j)^2} + \frac{2mg_1}{z-b_j} + g_1^2 - g_2} \quad (10)$$

Consider the case of $m = 2$. As $z \rightarrow b_j$ equation (10) essentially reduces to

$$\frac{P'''(z)}{P''(z)} \approx \frac{\frac{3m(m-1)g_1}{(z-b_j)^2}}{\frac{m(m-1)}{(z-b_j)^2}} = 3g_1 \quad (11)$$

As $z \rightarrow b_j$ for $m = 2$, the third derivative ratio approaches $3g_1$. If $m > 2$, $P'''(z)/P''(z) \approx (m-2)/(z-b_j)$ as $z \rightarrow b_j$.

Analysis of higher derivative ratios confirms the pattern indicated previously. The following generalizations can be made as z approaches b_j :

- (1) The constant approached by the $m+1$ derivative ratio is $(m+1)g_1$.
- (2) $P^{k+1}(z)/P^k(z) \rightarrow (m+1-k)/(z-b_j) \rightarrow 0$ for $1 \leq k \leq m$, so the number of derivative ratios that approach $\pm\infty$ is the number of roots m that are at b_j .
- (3) For $m > 1$ and an integer k in the range $1 \leq k \leq m$, $\left[\frac{|P^{k+1}(z)|}{|P^k(z)|} \right] / \left[\frac{|P^k(z)|}{|P^{k-1}(z)|} \right] \rightarrow (m-k)/(m+1-k)$ where the absolute value symbols mean the magnitude of the vector sum of the real and imaginary parts when $P(z)$ is complex.

These generalizations show how to interpret derivative ratios for m and g_1 as a root is approached. When at a trial point near a root, however, it is not easy to tell if a derivative ratio is approaching a constant or infinity. The ratio of successive derivative ratios, as partially introduced in generalization (3), is useful for this purpose. For a general integer k let us call this ratio ARATIO as it is in the computer program described in a later section of the report; then,

$$\text{ARATIO} = \frac{\frac{P^{k+1}(z)}{P^k(z)}}{\frac{P^k(z)}{P^{k-1}(z)}} \quad (12)$$

From generalizations (1) and (2) when $k = m$,

$$\text{ARATIO} = \frac{\frac{P^{m+1}(z)}{P^m(z)}}{\frac{P^m(z)}{P^{m-1}(z)}} \rightarrow (m+1)g_1(z - b_j) \rightarrow 0 \quad \text{as} \quad z \rightarrow b_j$$

From generalization (3), $\text{ARATIO} \rightarrow (m-k)/(m+1-k)$ for $1 \leq k < m$; thus, when k is a positive integer less than m , $|\text{ARATIO}| \rightarrow C$ where the constant lies in the range $0.5 \leq C < 1$ and when $k = m$, $|\text{ARATIO}| \rightarrow 0$. At a trial point in the vicinity of a root, the zero may not be very distinct; but any $|\text{ARATIO}|$ value less than 0.5 is an indication that the ARATIO associated with the particular k is headed for zero. The particular k can then be used as the current value of m .

The values of m and g_1 obtained from a trial point provide the means of calculating $z - b_j$ in equation (6) for the general case of several roots at a point. However, a more direct way is by equation (13)

$$z - b_j = \frac{1}{\left[\frac{P^m(z)}{P^{m-1}(z)} \right]} \quad (13)$$

where m is both the order of derivative and the multiplicity of the root. An excellent characteristic of these calculated $z - b_j$ values is that they become increasingly accurate as a root is approached. They are a powerful aid in converging to a root and in establishing a very accurate value for a root. The reason for better $z - b_j$ values as z

approaches b_j is the major terms in the ratio of derivative equations become increasing orders of magnitude larger than the terms ignored.

Limitations in Practical Applications

The preceding theoretical observations are useful only to the extent that they can be applied within the limitations encountered in practical work. For finding roots of polynomials the limitations are not severe; but they do exist; and they merit discussion. Almost all of the limitations are a result of the number of significant figures that can be carried for a constant or variable on the computer.

The basic constants and initial parameter values that are input to the computer have a round off error in the last significant figure. As mathematical manipulations are made on the computer these round off errors and other process errors make the probable relative error of calculated parameters, such as, $P(z)$ and its derivatives, larger. For meaningful ratio of derivative analysis it is necessary to recognize when the error of a computed value can be as large as the parameter itself. A relative error criterion can be established for the polynomial derivatives from an error analysis study.

The number of significant figures that can be carried on a computer and the relative error criterion in essence establish the maximum size of a single meaningful derivative ratio. However, the judgment on the multiplicity of a trail root is made with ARATIO which has two derivative ratios. Thus, it is necessary to have two reasonably accurate derivative ratios. Since, as indicated in the earlier ARATIO discussion, both of these derivative ratios may be approaching infinity, the maximum allowable size of a derivative ratio for the purpose of determining m is about the square root of the maximum size of a single meaningful derivative ratio; that is, about one-half the meaningful significant figures of a calculated derivative. This limit on the magnitude of a derivative ratio for the determination of m in essence establishes the minimum distance for which a computer program can resolve nearby roots rather than treat them as a multiple root. As indicated by equation (13), this minimum resolution distance for a nearby pair of roots is the inverse of the derivative ratio.

When a pair of root points are resolved, the error of the root point b_j is of the order of magnitude of the resolution distance. Usually the error of b_j can be reduced several orders of magnitude by using equation (13) for one more iteration to obtain b_j with the m established in resolution. The least improvement in accuracy is made if a pair of roots are the resolution distance apart. If a pair of roots are greater than the resolution distance apart, the order of magnitude of root location error reduction is the ratio of resolution distance to the distance between the root pair. If a root pair is less than the resolution distance apart, they are treated as a double root at the centroid of the root pair.

Whereas a pair of nearby roots can be resolved to a known accuracy, the resolution of clusters of nearby roots cannot be described as precisely. The approximate resolution distance of an evenly spaced group of roots m on a circle in the complex plane is the number of meaningful significant figures of a computed derivative divided by m . For example, if a computer has 16-significant-figure capability, it may be possible to retain about 14 significant figures in a polynomial derivative value of a tenth-degree polynomial. With four evenly spaced roots, the resolution distance would be only three and a fraction significant figures. The ratio of derivatives method, however, is most useful when closely packed clusters of roots or a multiroot point is encountered. In the approach to such a group of roots the polynomial appears to approach a high ordered zero or multiple root; so the actual value $P(z)$ stays well below the absolute error associated with a computed value of $P(z)$ for a range of z . In the case of an actual multiroot point each of the $m - 1$ derivatives approach a lower order zero. Thus, it may not be possible to evaluate $P(z)$ and its lower derivatives, but it works out nicely that the derivative ratios needed for root resolution (determination of m) are the ones that can be calculated accurately. In fact, the advantage of the derivative ratio method over other methods is that root analysis can still be done even though the polynomial and its lower order derivatives cannot be evaluated with sufficient accuracy. Upon near range approach to a cluster, individual roots can usually be resolved; but in the cases where resolution cannot be made the remaining group is treated as a multiroot located near the centroid of the group.

Summary of the Algorithm

The major features of the ratios of derivative method have been discussed at some length. In the following stepwise procedure the ideas are summarized as they might be used to find roots of polynomials:

(1) Find a trial z for which $P(z)$ is in the vicinity of zero. The ratio of derivatives method usually works for this, but it may not be either efficient enough or reliable enough for a general program. It may be advisable to use some standard form of two- or three-term Taylor's series expansion of the polynomial for this phase.

(2) When in the vicinity of a root evaluate the first derivative ratio, $P'(z)/P(z)$ to determine the approximate location of the root. $P'(z)/P(z)$ is an order of magnitude measure of the closeness of z to b_j .

(3) Find $P''(z)$ and calculate $P''(z)/P'(z)$. From equation (12), if $P''(z)/P'(z)$ is greater than one-half of $P'(z)/P(z)$, there is a multiple root or at least two roots within $1/[P'(z)/P(z)]$ of each other so that they cannot be resolved yet. Continue taking derivative ratios, $P^k(z)/P^{k-1}(z)$, until one is found which is less than one-half the next lower order one. The current multiplicity m of the root b_j is $k - 1$ where $P^k(z)/P^{k-1}(z)$

is the first derivative ratio that is less than one-half of the next lower order derivative ratio. As a group of roots is approached, it may be possible to resolve roots that looked like multiple roots from a distance; consequently, m may be lowered during the z trials.

(4) Adjust z with the following relation:

$$z_{\text{new}} = z - \frac{1}{\left[\frac{P^m(z)}{P^{m-1}(z)} \right]} \quad (14)$$

As a root is more closely approached, this correction becomes better by orders of magnitude. If a multiroot point is approached in iterating, the values of $P(z)$ and lower order derivatives approach high order zeros. Thus, it may not be possible to get values for them, but the higher derivatives can be evaluated for m and the z adjustment.

(5) Locate z within a tolerance of about one-half the order of magnitude of the root resolution criterion. For the purpose of resolving nearby roots, two successive derivative ratios which are free of round off or truncation errors are needed. Thus, a root resolution criterion of about one-half the significant figures that can be retained in a calculated derivative ratio should be used. If a z value is apparently closer to b_j than the criterion range, it should be backed away until in the criterion range.

(6) After z is in the root-resolution criterion range, calculate g_1 from

$$g_1 = \frac{1}{m+1} \frac{P^{m+1}(z)}{P^m(z)} \quad (15)$$

and improve the value of b_j . When $P^m(z)/P^{m+1}(z)$ is free of round off or truncation error, the error in b_j can usually be reduced by several orders of magnitude with

$$b_j = z - \frac{1}{\left[\frac{P^m(z)}{P^{m-1}(z)} \right]} \quad (16)$$

(7) Divide the roots out of the polynomial. The order of the polynomial will be reduced from n to $n - m$.

(8) Estimate the location of the next root. The value of g_1 is very nearly

$$g_1 \approx \frac{1}{z - b_1} + \frac{1}{z - b_2} + \dots + \frac{1}{z - b_{n-m}} \quad (17)$$

If another root is relatively close to the root just found, one term in equation (17) should predominate. A rough approximation of the initial b_j can then be

$$b_j \approx z - \frac{1}{g_1} \quad (18)$$

These general steps were used in a computer program for finding the roots of polynomials with real coefficients. A computer program for only real number polynomials rather than the general case of complex number polynomials is discussed for three reasons. First, far more real number polynomials are used in practical applications. Second, a computer program specifically constructed for only real number polynomials requires somewhat fewer computer operations and, thus, is more efficient for the bulk of problems. And third, discussion of a real number polynomial program may more fully illustrate the application of the ratio of derivatives concept since almost all of the logic needed for the general case, plus that specifically for real roots, is used.

DESCRIPTION OF THE COMPUTER PROGRAM

From both accuracy and efficiency considerations it is advantageous to structure a program to do as much analysis in real number algebra as possible. The reason is that fewer computer operations are required for real number computations than for complex number computations. Thus, this program makes a thorough search for all real roots first; so that the polynomials are often reduced in degree, and hence length, before complex number algebra is needed. The program can be considered to be composed of two parts, one with real algebra operations and the other with complex number operations. The major features of each are discussed.

Program Segments Coded in Real Number Algebra

Most of the real number operations are done in the main program, ROOTS, which also serves as the control routine. The remaining real number operations are done in subroutines which are mentioned by name at appropriate places in the discussion.

Preliminary calculations. - At first a check of the input data with the dimension limits is made. Then some tests for easy reduction of polynomial degree are made. If the highest degree coefficient is zero, the polynomial degree is lowered by one. The test is repeated until a nonzero coefficient is found. The same type of test also is made at the low degree end of the polynomial. A root value of zero is associated with each

zero coefficient on the low degree end. Thus, the roots equal to zero are immediately accounted for, and the polynomial degree is reduced without further ado.

One other calculation of a preliminary nature is a possible gross scaling of the polynomial. Scaling makes the root resolution criterion a fraction of average root size instead of a fixed absolute value. The scaling is done by hexadecimal orders of magnitude. Hexadecimal scaling is used since no additional error is introduced into the polynomial coefficients with computer multiplications and divisions by 16 on present computers. A scaling decision is made from a least-squares line fit of hexadecimal logarithms of the polynomial coefficients. If this line has a slope between +0.5 and -0.5, no gross scaling is done.

Polynomial and polynomial derivative calculation procedure. - The real number polynomial can be written as

$$P(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1} \quad (19)$$

where the a 's are now real numbers indexed from the high degree end of the polynomial. Indexing the polynomial coefficients in this manner conforms to the order they are input in the program.

To qualitatively judge the nearness of a polynomial to a root, it is desirable to normalize equation (19). Thus, it could be rewritten as

$$P(x) = \left(\left[\left(\frac{a_1}{a_{n+1}}x + \frac{a_2}{a_{n+1}} \right)x + \frac{a_3}{a_{n+1}} \right]x + \dots + \frac{a_n}{a_{n+1}} \right)x + 1 \Big) a_{n+1} \quad (20)$$

where the term in braces is normalized to one. As a root is approached, the term in braces approaches zero; but its value is meaningful only when above the absolute error for the calculated value of $P(x)$. The absolute error of the term in braces, of course, grows with degree of the polynomial. For a tenth-degree polynomial the absolute error can be expected to be approximately two orders of magnitude greater than the absolute error of the same input number. If the first roots found are somewhat isolated, their relative error will be almost as good as that of $P(x)$. In the process of dividing out a root, additional error is introduced into the remaining polynomial coefficients. Thus, even though the polynomial gets shorter, the last roots, in general, have greater relative error. When the roots are all relatively isolated from each other, the root values are certainly accurate enough for engineering applications (see examples in appendix C).

However, when several roots lie at a point or are clustered, the resolution is not as good as with isolated roots; consequently, the accuracy of such roots is not as good either. Unfortunately, when polynomials have multiple roots or clusters of roots, the relative error of $P(x)$ usually is higher too. The reason is believed to be a function of

the relative size of the polynomial coefficients. For illustration, note that a polynomial of multiple roots can be written as $(x - a)^n$. As n increases the center coefficients of the binomial series expansion of $(x - a)^n$ becomes orders of magnitude larger than the end coefficients. Another difficulty with binomial series coefficient polynomials is there are terms nearly equal in magnitude but of opposite size. Consider, for example, $(x - 1)^5 = x^5 - 5x^4 + 10x^3 - 10x^2 + 5x - 1$. Note that as x approaches 1, $10x^3$ approaches $10x^2$ and the resulting sum will have few significant figures. Table I illustrates how small the value of this polynomial is in the vicinity of its multiple root.

For the purposes of illustration, table II shows how the finite precision of a computer makes it impossible to locate multiple roots accurately if the polynomial equals zero criterion is used. Even using ten significant figures in the evaluation of the polynomial, the value of $x = 0.999$ will be accepted as a root. Thus, a ten-significant-figure calculation does not even yield a three-significant-figure root. Furthermore,

TABLE I. - VALUES OF THE
POLYNOMIAL $(x - 1)^5$ AT
VARIOUS x

$$[P(x) = (x - 1)^5 = x^5 - 5x^4 + 10x^3 - 10x^2 + 5x - 1]$$

x	P(x)
0.8	0.00032
.9	.00001
.99	.000 000 449
.999	.000 000 000 000 001

TABLE II. - EVALUATION OF THE POLYNOMIAL

$$(x - 1)^5 \text{ at } x = 0.999$$

(a) Using infinite precision

n	A_n	x^n	$A_n x^n$
0	-1	1.0	-1.0
1	5	.999	4.995
2	-10	.998 001	-9.980 01
3	10	.997 002 999	9.970 029 99
4	-5	.996 005 996 001	-4.980 029 980 005
5	1	.995 009 990 004 999	.995 009 990 004 999
$P(x) = \sum_{n=0}^5 A_n x^n = -0.000\ 000\ 000\ 000\ 001$			

(b) Truncating to ten significant figures

n	A_n	x^n	$A_n x^n$
0	-1	1.0	-1.0
1	5	.999	4.995
2	-10	.998 001	-9.980 01
3	10	.997 002 999	9.970 029 99
4	-5	.996 005 996 0	-4.980 029 980
5	1	.995 009 999 00	.995 009 990 0
$P(x) = \sum_{n=0}^5 A_n x^n = 0.000\ 000\ 000\ 0$			

after dividing out the root, the coefficients of the reduced polynomial will have at most three significant figures.

Again for illustration the following table shows the buildup of error as each new root is found with less accuracy than the preceding root:

$x_1 = 1.0063$
$x_2 = 0.9985 + 0.0061 i$
$x_3 = 0.9985 - 0.0061 i$
$x_4 = -4.9015 - 0.7623 i$
$x_5 = -9.4521$

The polynomial, $(x - 1)^5 = x^5 - 5x^4 + 10x^3 - 10x^2 + 5x - 1$, was input to a computer program using Laguerre's method for finding roots. The program utilizing eight significant figures in its calculation found the first root to about three significant figures. The next two roots were also located to about three significant figures but were reported to be complex instead of real. The last two roots are not even accurate to within one significant figure. However, the accuracy and internal checks within the program were satisfied and the program reported no error message indicating the roots were bad. The only indication of trouble is the appearance of a complex root without a conjugate.

Even with the derivative ratio program, it is better to use the most accurate calculation procedure known for the calculation of $P(x)$ to retain as much resolution capability as possible.

An alternate method of evaluating $P(x)$ is the following form:

$$P(x) = \left[\left(\dots \left\{ \left[\left(\frac{a_1}{a_2} x + 1 \right) \frac{a_2}{a_3} x + 1 \right] \frac{a_3}{a_4} x + 1 \right\} \dots \right) \frac{a_n}{a_{n+1}} x + 1 \right] a_{n+1} \quad (21)$$

The epsilon, delta error analysis method of reference 3 applied to the previous forms for evaluating the polynomial $P(x)$ shows that the latter form is probably the more accurate when clusters of roots are encountered, even though there are more operations used. Part of the reason is that no additional absolute error is introduced in the additions of the exact constant one. For a fifth-degree root the relative errors of $P(x)$ computations by equations (20) and (21) are the same; but for a tenth-degree root the relative error of $P(x)$ computed by equation (21) is about 5 percent less than that of equation (20). There is some question as to whether this is enough difference to warrant the use of the

latter form; but it was used in the program since the overall program is generally quite fast and efficient.

Equation (21) cannot be used directly, however, because a zero coefficient gives the division-by-zero problem. This problem can be handled, in general, by checking the magnitude of each coefficient with some tolerance which should be about the absolute error of other polynomial coefficients. If some polynomial coefficients a_j 's are below the tolerance, the principle illustrated by equation (22) can be used to evaluate $P(x)$ in the following way:

$$P(x) = \left\{ \left[\dots \left(\left\{ \dots \left[\left(\frac{a_1}{a_2} x + 1 \right) \frac{a_2}{a_3} x + 1 \right] \dots \right\} \frac{a_{j-1}}{a_{j+1}} x^2 + 1 \right) \dots \right] \frac{a_n}{a_{n+1}} + 1 \right\}^{a_{n+1}} \quad (22)$$

Derivatives of $P(x)$ also are polynomials, so the preceding equation forms are applicable for them too. Calculation of the derivatives is simply a matter of setting up the derivative coefficients and adjusting the degree of the polynomial. The calculation of a polynomial and its derivatives, with x the only independent variable, is the specific function of subroutine RPOLY. At any point x during iteration $P(x)$, $P'(x)$, and $P''(x)$ always are calculated. Higher order derivatives are calculated only if the ratio of derivatives segment of the program has a need for them.

Approximate root location with second order Taylor's series. - A second order Taylor's series is used for moving x in the vicinity of a root. This method was chosen because, first, it is quite efficient in moving toward roots and, second, it can be programmed to almost certainly get sufficiently near all the real roots.

The series may be written as

$$P(x + h) = P(x) + hP'(x) + \frac{h^2}{2} P''(x) \quad (23)$$

Since real roots are the x values for which $P(x)$ is zero, the x increment, or h , which makes $P(x + h)$ equal zero is sought. With h the only unknown in equation (23), the quadratic equation can be solved for values of h . If the h 's are real, the polynomial appears headed for a root. The value of x then is corrected by the h with the smaller magnitude. And $P(x)$ and its derivatives are recalculated with the new x . When the absolute value of $P(x)$ gets below the resolution distance TOLR, the ratio of derivatives procedure is entered.

If the h 's in the Taylor's series quadratic equation are not real, the polynomial either has made an approach and retreat from the x -axis or appears to be making an approach and retreat from the x -axis. Through logical use of $P(x)$, $P'(x)$ and their

comparison with $P(x)$ and $P'(x)$ from the previous x trial, the polynomial can be investigated in this region to readily determine if this is indeed an x -axis approach and retreat rather than a root. If the region is only an approach, increasing x -increment steps are taken away from the region in the direction started. These steps are continued until either another real h is found or 15 steps have been taken in which case x values are investigated in the other direction from the starting point. If no real roots are found in that direction either, the complex number analysis subroutine, COMPLX, is called.

The starting point in the search for the first real root is $x = 0$. The first x movement is in the direction of decreasing $|P(x)|$. After a root has been found, the starting point in the search for the next root is the estimated x value from the ratio of derivatives analysis.

Real number analysis by ratio of derivatives. - The most important parameter in the program for making decisions with ratio of derivatives analysis is ARATIO, which is defined as

$$ARATIO = \frac{\left| \frac{P^{k+1}(x)}{P^k(x)} \right|}{\left| \frac{P^k(x)}{P^{k-1}(x)} \right|} \quad (24)$$

where k is the order of a polynomial derivative. This parameter is useful because we know from theory that

$$ARATIO \ll 0.5 \quad \text{for} \quad k = m \quad (25)$$

and

$$1.0 > ARATIO \geq 0.5 \quad \text{for} \quad 1 \leq k < m \quad (26)$$

as a root is approached. In the program 0.45 was used for the 0.5 in equations (25) and (26) to give a little allowance for error. These constants in essence establish when roots are treated as multiple or close together.

The remaining possibility of ARATIO greater than one is an indication of either an x value between or among roots but not relatively near any or an encounter with absolute error of a computed derivative rather than a true value. To do meaningful analysis, it is necessary to know which it is. So it is important to establish a good polynomial error criterion. The criterion parameter in the program is TOLRQ. It is the probable absolute error of a computed polynomial value normalized by the lowest degree constant

which is nonzero. The normalized polynomial or polynomial derivative in the program is $P(ND)$; and it is to be distinguished from $POLY(ND)$, the value of the polynomial or polynomial derivative. Also note that since a zero subscript cannot be used in some computer languages, $P(1)$ denotes the normalized polynomial, $P(2)$ is the normalized first derivative, and so forth; that is, $ND = k + 1$.

$TOLRQ$ should be greater than the probable absolute error of $P(ND)$; but excess margin cuts into the root resolution distance criterion $TOLR$, which is the square root of $TOLRQ$. Consequently, $TOLRQ$ should be set with care. In this program $TOLRQ$ varies with the degree of the polynomial as a result of an epsilon, delta error analysis of multiroot polynomials or varying degree. After the functional relation was established, the magnitude of $TOLRQ$ was finally set by noting when computed values became different from known polynomial values for specific cases. With this functional relation of $TOLRQ$, better root accuracy can be expected with low degree polynomials.

The resolution criterion coefficients are essentially an absolute distance. However, it is probably desirable to have the resolution criterion be a relative distance to the local root. This effectively can be accomplished by scaling the polynomial so the local root is of order one. To accomplish this, subroutine $SCALER$ is called for an order of magnitude check of x on only the first pass through the ratio of derivatives analysis of each root. If the hexadecimal logarithm of x is not within ± 0.5 of zero, the polynomial will be hexadecimally scaled to bring the local x to order one for root analysis. However, since the polynomial coefficients must be held within the exponent limits of a computer, scaling in some cases is limited.

In recognition of the probable magnitude of polynomial absolute error, an attempt is made to begin the ratio of derivatives analysis with $P(ND - 1)$, $P(ND)$, and $P(ND + 1)$ values which have magnitudes greater than $TOLRQ$. Initially, ND is two so $P(1)$, $P(2)$, and $P(3)$ are calculated. The procedure is to then check $P(1)$ with $TOLRQ$. If $|P(1)| < TOLRQ$, ND is increased by one if ND is less than the degree of the polynomial. Then $|P(ND - 1)|$ is checked with $TOLRQ$. The procedure is repeated until $|P(ND - 1)|$ is greater than $TOLRQ$ or ND equals the degree of the polynomial. If $|P(ND - 1)| > TOLRQ$, the logical parameter $LIMIT$ is set to zero in the program to indicate complete ratio of derivative analysis is possible.

$ARATIO$ is computed from the three polynomial and/or polynomial derivative values, $POLY(ND - 1)$, $POLY(ND)$, and $POLY(ND + 1)$. When $LIMIT = 0$ and $ARATIO < 1$, analysis is done according to theory. When $LIMIT = 1$, the computed value of $POLY(ND - 1)$ is expected to have an absolute error value rather than an appropriate value for $POLY(ND - 1)$. Thus, the analysis methods based on an accurate value of $POLY(ND - 1)$ are not used. However, some of the ratio of derivatives judgments can still be made on the assumption that the actual value of $|POLY(ND - 1)|$ is probably less than the computed value of $|POLY(ND - 1)|$ since the lowest allowable

program limit of $|P(ND - 1)|$ is $0.5 * TOLRQ$.

When $ARATIO > 1$ and $LIMIT = 0$, the present x value is between or among roots. Some knowledge of the nature of $ARATIO$ contours about roots is helpful in establishing procedures when $ARATIO$ is greater than one. As an example, the $ARATIO$ contours for the polynomial, $P(z) = z^2(z - 1)(z - 100)$, are shown on the complex plane in figures 1 to 3. Such plots for known problems yield a great deal of information as to how to handle general cases. In this example, note that, for the group of three real roots, (1) the relative symmetry, (2) the points where $ARATIO$ approaches zero and infinity, and (3) the values of $ARATIO$ at derivative ratios above and below the point where an $ARATIO$ approaches zero or infinity. From these figures for this single problem the following elements of procedure evolved:

(1) When $ARATIO$ for ND is greater than one, step in a direction until an $ARATIO$ very near zero is found.

(2) This low $ARATIO$ may be a false root or roots, so check the root candidate as follows:

(a) If $ND = 2$, either $P(x)$ or $P''(x)$ approaches zero; so if $|P(x)| < |P''(x)|$, x is the root.

(b) If $ND > 2$, $ARATIO$ for $ND - 1$ at a root should be 0.5 by the theory.

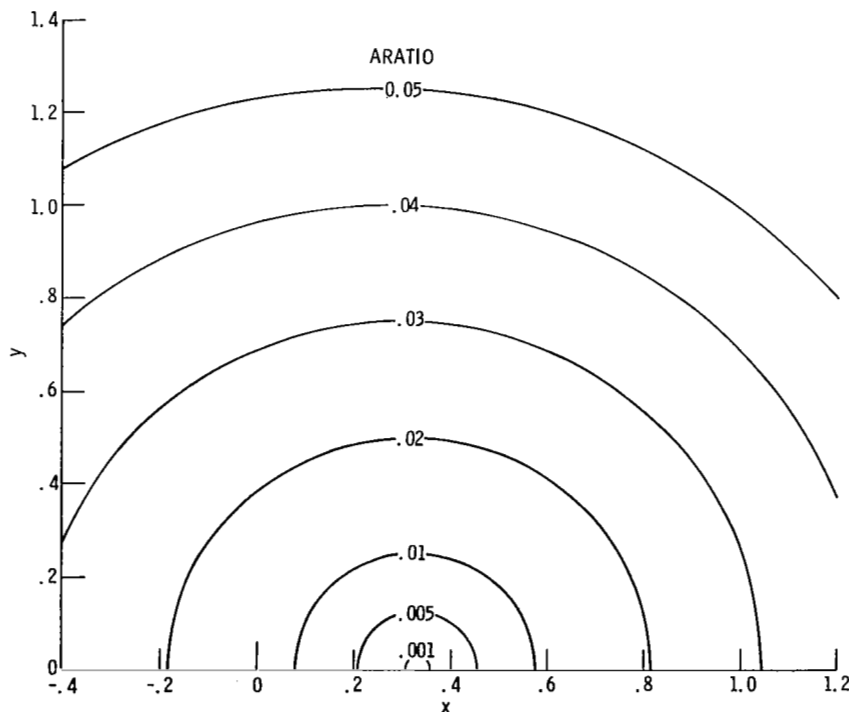


Figure 1. - Third degree $ARATIO$ contours for polynomial $P(z) = z^2(z - 1)(z - 100)$ where

$$ARATIO = \frac{|P''''(z)|}{|P'''(z)|} \bigg/ \frac{|P'''(z)|}{|P''(z)|}$$

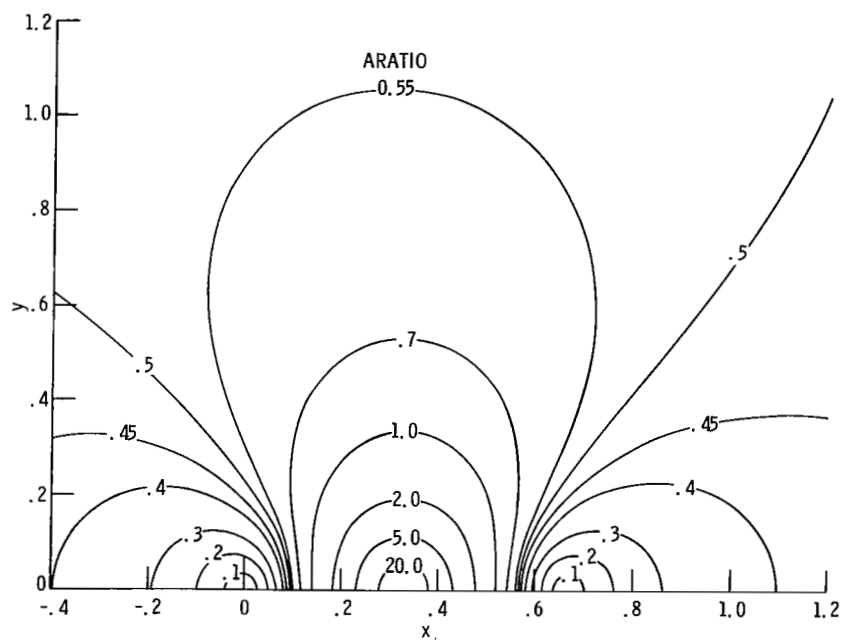


Figure 2. - Second degree ARATIO contours for polynomial $P(z) = z^2(z - 1)(z - 100)$ where

$$\text{ARATIO} = \frac{|P'''(z)|}{|P''(z)|} \bigg/ \frac{|P''(z)|}{|P'(z)|}$$

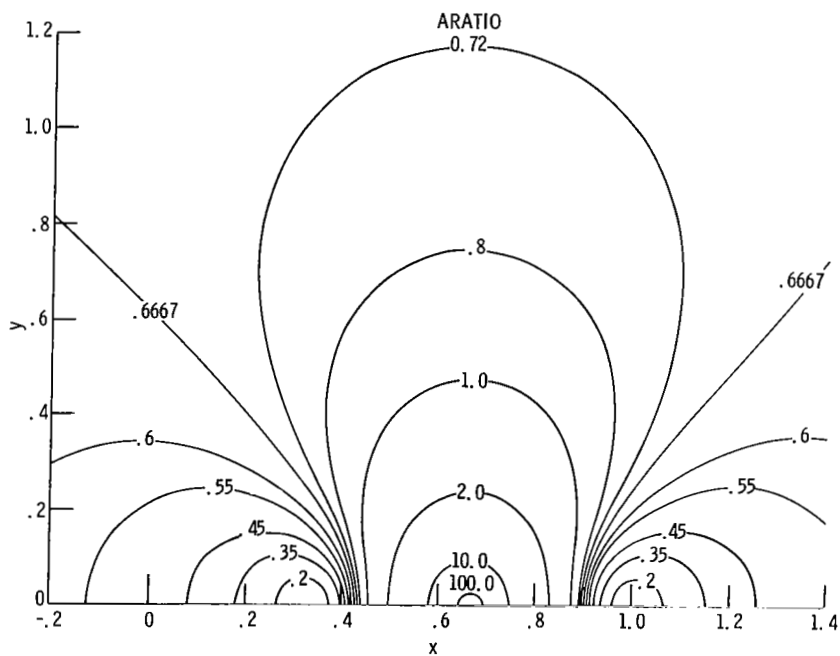


Figure 3. - First degree ARATIO contours for polynomial $P(z) = z^2(z - 1)(z - 100)$ where

$$\text{ARATIO} = \frac{|P''(z)|}{|P'(z)|} \bigg/ \frac{|P'(z)|}{|P(z)|}$$

(3) If the root point is false (i. e., $ARATIO$ for $ND - 1$ approaches infinity when $ND > 2$ or $|P''(x)| < |P(x)|$ when $ND = 2$), set x at the opposite point of symmetry and make the root test again.

A summary of the basic logical steps for the ratio of derivatives analysis for real roots is given in appendix D. The general philosophy of resolving roots in a group is to start with an ND equal to $m + 1$ and work down in ND as root resolution progresses. The reason is that multiple roots should be found first. The multiple root analysis is done with higher derivatives, which often can be calculated accurately, whereas the lower derivatives and $P(1)$ may be below $TOLRQ$, so that the single root analysis cannot be done accurately.

If upon entry of the ratio of derivatives part of the program the x point is outside the group by a distance of approximately one or two times the maximum distance between roots in the group, the group will appear at first to be a multiple root; so the initial ND value naturally will be raised to the number of roots in the group plus one (note that the values of $ARATIO$ away from roots in figs. 2 and 3 are always greater than 0.5). However, if the first x trial in the ratios of derivative analysis is within the group, it is probable that a low $ARATIO$ of a false root will be found at a lower than desired ND . The test for this situation is to raise ND by one and check the $ARATIO$ at the higher ND . If this $ARATIO > 1$, analysis is begun at this higher ND ; but if $ARATIO < 1$, analysis is begun at the original ND .

The aforementioned procedures probably would work quite well for only real roots; but a program must be able to handle the cases of real and complex roots in a group. $ARATIO$ contours for known problems of this type show definite patterns; but the contours are not as distinctive as those for only real roots. These contours, however, give clues as to how to search for a real root when one is known to exist in the group.

An odd number of real roots is known to exist within a range of x if $P(1)$ is known to change sign in that range. As an aid for the analysis of a group of roots a running record of a possible root range is kept. If values of $P(1)$ distinguishable from the absolute error are known to change sign between x trials, the logical parameter $IMSURE$ is set to one. At each new x trial throughout the Taylor's series investigation and ratio of derivative analysis, $P(1)$ is checked; so the root range is often narrowed with each new x trial. If $IMSURE$ is one, the root group $ARATIO$'s will be quite carefully searched for a real root over the known root range. It may not be possible to find a root because the necessary $P(ND)$ values are below $TOLRQ$. In this case x values are found which place $|P(1)|$ values at the ends of the root range between 0.5 and 1.0 times $TOLRQ$; and a root is divided out at midrange x .

Dividing out roots. - After a root or multiple roots are identified, the normal procedure is to (1) locate x within the range of $0.2 * TOLR$ and $1.0 * TOLR$ of b_j , (2) estimate the location of the next root, (3) improve the value of the present root, and

(4) divide the root out of the polynomial. The x estimate for the next root is given by the following equation:

$$x_{\text{next}} = x - \frac{m+1}{\left[\frac{P^{m+1}(x)}{P^m(x)} \right]} \quad (27)$$

If $|P(\text{ND} - 1)| > \text{TOLRQ}$, the value of the present root can be improved, usually by several orders of magnitudes, by the following equation:

$$x_r = x - \frac{1}{\left[\frac{P^m(x)}{P^{m-1}(x)} \right]} = x - \frac{1}{\left[\frac{\text{POLY}(\text{ND} + 1)}{\text{POLY}(\text{ND})} \right]} \quad (28)$$

The number of roots m found are divided out of the polynomial one at a time by the method presented on pages 76 and 77 of reference 3. This method is shown in appendix B for convenience. The normalized polynomial remainder (polynomial relative error) left after each root is divided out is also saved for output. The polynomial remainder should be zero; so its value is a good clue as to whether or not the root was determined as accurately as it should have been.

When the degree of the polynomial is reduced to two or less, subroutine QED is called. Any remaining roots are found by direct computation. Then all of the polynomial roots and remainders are printed.

Program Segments in Coded in Complex Number Algebra

If more than two roots remain after the search for real roots has been completed, subroutine COMPLX is called for the search of complex conjugate pairs of roots. The procedures in COMPLX somewhat parallels those of ROOTS.

Polynomial and polynomial derivative calculation procedure. - The polynomial $P(z)$ and its derivatives are calculated with a form of equation (21) for complex numbers. The calculation of a polynomial and its derivatives with z the independent variable is the specific function of subroutine CPOLY. At any z point during iteration, $P(z)$, $P'(z)$, and $P''(z)$ always are calculated. Higher order derivatives are calculated only if the ratio of derivatives segment of the program has a need for them.

Approximate root location with second order Taylor's series. - A second order

complex Taylor's series is used to locate z in the vicinity of a root. The series may be written as

$$P(z_r) = P(z) + P'(z)(z_r - z) + P''(z) \frac{(z_r - z)^2}{2} \quad (29)$$

Since roots are z_r values from which $P(z_r)$ is zero, equation (29) can be written as

$$0 = P(x) + iP(y) + [P'(x) + iP'(y)](z_r - z) + [P''(x) + iP''(y)] \frac{(z_r - z)^2}{2} \quad (30)$$

The solution of equation (30) for $z_r - z$ can be expressed as

$$z_r - z = \frac{-[P'(x) + iP'(y)]}{P''(x) + iP''(y)} \pm \frac{\sqrt{[P'(x)]^2 - [P'(y)]^2 - 2[P(x)P''(x) - P(y)P''(y)] + 2i[P'(x)P'(y) - P(y)P''(x) - P(x)P''(y)]}}{P''(x) + iP''(y)} \quad (31)$$

The plus or minus sign on the square root term of equation (31) indicates that the term may be used as computed or at a 180° phase angle. Both solutions for $z_r - z$ are computed. The one that gives the smaller absolute value of $z_r - z$ in the complex plane is used for the new trial z_r . When the absolute value of $P(z_r)$ is reduced below the resolution tolerance TOLI, the ratio of derivatives analysis is begun.

The initial coordinates of the Taylor's series search for the first complex root are the x that corresponds to some local minimum of $|P(x)|$ in the real number Taylor's series analysis in ROOTS and a y value equal to the square root of $|P(x)|$. After a complex conjugate pair of roots have been found, the initial coordinates in the search for the next roots are the estimates from the final stages of the ratio of derivatives analysis of the present root.

Complex number root analysis by ratio of derivatives. - It is rather interesting that all of the information extracted from real number ratio of derivative analysis along the x -axis can also be extracted from complex number ratio of derivative analysis in the complex plane. The reason for this lies with the very definition of analytic complex number derivatives. The essential point is that a derivative has the same value at a point no matter from which direction the point is approached. The complex derivatives have real and complex parts which can be vectorially combined into derivative magni-

tudes. Ratio of derivative analysis based on these derivative magnitudes (ARATIO'S) works in the same way as real number ratio of derivative analysis for the determination of m .

As with real roots the derivative ratios used to establish m are the ones from which the component adjustments are made. However, in complex numbers it is necessary to break down the appropriate derivative ratios into real and imaginary parts to get x and y adjustment values. Since the development of appropriate equations is somewhat complicated, it is done in appendix B, and only the results are shown here. The adjusted or new x and y values are obtained from the derivatives by the following equations:

$$x_{\text{new}} = x - \frac{A}{A^2 + B^2} \quad (\text{B32})$$

$$y_{\text{new}} = y + \frac{B}{A^2 + B^2} \quad (\text{B33})$$

where

$$A = \frac{P^m(x)P^{m-1}(x) + P^m(y)P^{m-1}(y)}{[P^{m-1}(x)]^2 + [P^{m-1}(y)]^2} \quad (\text{B27})$$

and

$$B = \frac{P^m(y)P^{m-1}(x) - P^m(x)P^{m-1}(y)}{[P^{m-1}(x)]^2 + [P^{m-1}(y)]^2} \quad (\text{B28})$$

In the ratio of derivatives analysis, the logical parameter LIMIT is again used to indicate whether or not the computed polynomial value is greater than the probable absolute error. The complex number criterion TOLIQ is of the same form as TOLRQ; but TOLIQ was set at four times TOLRQ. This is because more computer operations are needed to compute a complex number polynomial value than for a real number polynomial of the same degree. The complex root resolution criterion TOLI is the square root of TOLIQ.

For the initial z trial in ratio of derivatives analysis two checks are made. First, the degree of the root approached is raised as necessary within the restraints of the degree of the polynomial to begin the analysis with LIMIT equal to zero if possible. And

second, the polynomial is scaled if the magnitude of z is not sufficiently near one.

If, during ratio of derivatives analysis, ARATIO is less than one and LIMIT is zero, the analysis follows the theory. When ARATIO is greater than one with LIMIT equal to zero, the present point lies between or among roots. Since there is another adjustment degree of freedom on a plane as compared to that along a line, there is greater desire to have z movement procedures which effectively and efficiently move toward roots. Again the ARATIO contours in the complex plane give clues of effective procedures. See figure 4 for an example of contours about complex roots. Also, the

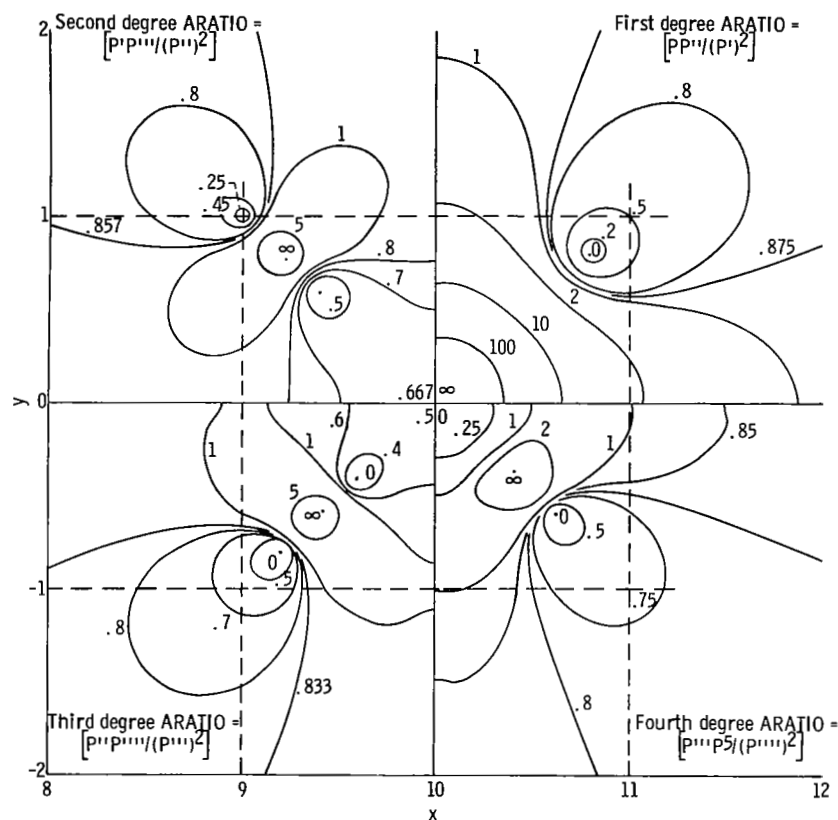


Figure 4. - ARATIO contours for polynomial $P(z) = (z - 9 + i)^2(z - 9 - i)^2(z - 11 + i)^2(z - 11 - i)^2$.
(Note: The polynomial is symmetric in four quadrants about the point $x = 10, y = 0$.)

ARATIO contours of figures 1 to 3 are quite similar to those in the vicinity of a similar grouping of complex roots. When ARATIO is somewhat greater than one, the contours are nearly circles. Clearly steps should be toward lower ARATIO; but a direction started may not necessarily head in the direction of an ARATIO less than 0.45. Consequently, the need of a curved path is indicated.

In the program the procedure, when ARATIO is greater than one, is to test step

in the x direction and then in the y direction. From these two normal steps the most effective direction for reducing ARATIO is determined. Then a spiral path in 30° increments is followed in the search for an ARATIO less than 0.45. If while on the spiral ARATIO increases from the previous iteration, analysis reverts back to the previous point. At that point, a new direction is determined from another set of x and y test steps; and then a spiral turning in the opposite direction is begun. The experience is that the program should seldom have to move more than seven points on any one spiral or should seldom have to begin a new spiral more than once.

In the complex plane the general philosophy of resolving roots in a group is nearly the same as on the real axis in ROOTS. There are some differences, however, which deserve discussion. First, since there is freedom to move about the complex plane, the problem of resolving complex from real roots no longer exists. If it turns out that there still are real roots in the group, they can be identified in the complex plane too.

A second difference is in the check of the degree of the root candidate investigated. In ROOTS a root candidate was quite well located before a final check of the root degree (ND level) was made. In the complex plane there are not quite as many logical parameter possibilities, so it is a little easier to change the root degree at any time during the ratio of derivative analysis. This is done by checking ARATIO and another ARATIO at the next lower ND for each z trial when the degree of the root is greater than one.

A third difference is that the symmetry of ARATIO contours cannot be depended upon. In a root group it is less probable that points where ARATIO equals zero will be symmetrical with each other about a point where ARATIO equals infinity. However, by keeping track of ARATIO's at the two ND's it is usually possible to identify false roots quickly.

Dividing out roots. - As a root or multiple roots are approached, the final decision on the multiplicity of the root is made when z is within the range of $0.2 * TOL$ and $1.0 * TOL$ of b_j . After m is established, the location of the next nearest root is estimated from g_1 found by

$$g_1 = \frac{1}{m+1} \frac{P^{m+1}(z)}{P^m(z)} \quad (36)$$

Since the conjugate of the present root is also in g_1 , the breakdown of g_1 for the next nearest root location is a rather lengthy development which is shown in appendix B. The resulting equations for the new coordinates are

$$x_{\text{next}} = x_r - \frac{g_R y_r}{y_r \left[g_R^2 + \left(g_I + \frac{m}{2y_r} \right)^2 \right] + g_I + \frac{m}{2y_r}} \quad (37)$$

and

$$y_{\text{next}}^2 = \frac{\left[g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}}) \right] \left[(x_r - x_{\text{next}})^2 + y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}})} \quad (38)$$

where g_R and g_I , respectively, are the real and imaginary parts of g_1 . The values of the present root coordinates then are improved with the same equations used for coordinate adjustment, that is, equations (32) and (33).

A complex root is first artificially divided out of the polynomial to find the real and imaginary normalized polynomial remainders. Since these remainders should be zero, their values are clues as to whether or not the complex root was located as accurately as it should have been. The dividing out process is artificial in the sense that the polynomial coefficients are not permanently changed until the root and its conjugate are divided out together in real algebra. The equations and details for dividing out the complex roots by both methods are shown in appendix B. The polynomial remainder from the real algebra division appears in the same printout line with the conjugate root.

Examples

The general capabilities and limitations of the program have been discussed; but they are most effectively shown with examples. The exact roots of several polynomials that are difficult to solve are shown with the computed roots in the program output section of appendix C. These examples illustrate the root resolution capability of this program.

Appendix C also contains a listing of the program, a description list of the program variables, and other special instructions for a user.

CONCLUDING REMARKS

The ratio of derivatives method is a powerful method of finding roots of polynomials.

A computer program for finding the roots of real number polynomials by this method was developed to see how useful the theory is within the confines of calculated number accuracies. Examples of the root resolution power of this program illustrate that the method is indeed powerful in practice as well as in theory. Although a program was not developed for complex number polynomials coefficients, equally effective root finding programs can be developed with the ratio of derivatives method.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, March 2, 1972,
764-74.

APPENDIX A

SYMBOLS

A	combination of complex derivative terms (see eq. (B27))
a	general polynomial coefficient
B	combination of complex derivative terms (see eq. (B28))
b	general coordinate of roots
c	general polynomial coefficient after a root is divided out
f_1	function of the form $\sum_{j=1}^n [1/(z - b_j)]$
f_k	function of the form $\sum_{j=1}^n [1/(z - b_j)^k]$
g_1	f_1 at a root with the local root point excluded, $\sum_{j=1}^{n-m} [1/(z - b_j)]$
g_k	f_k at a root with the local root point excluded, $\sum_{j=1}^{n-m} [1/(z - b_j)^k]$
h	x adjustment increment toward a root by Taylor's series
k	degree of an arbitrary polynomial derivative
l	arbitrary exponent of the function f_1 in eq. (B21)
m	number of roots at the particular root point
n	number of roots in the polynomial
P(x)	polynomial in real numbers
P(z)	polynomial in complex numbers
$P^k(z)$	k^{th} polynomial derivative in complex numbers
x	independent variable or real component of independent variable
y	imaginary component of independent variable
z	independent variable in complex number form
Subscripts:	
I	refers to imaginary part of a complex variable
j	an arbitrary term or root of a polynomial

n n^{th} and last root or next to last polynomial coefficient
 new newest approximation to a root
 next estimate of next root location
 R refers to real part of a complex variable
 r root or very near root value

Superscripts:

j arbitrary term in the polynomial
 k degree of an arbitrary polynomial derivative
 l arbitrary power of the function f_1 in eq. (B21)
 m number of roots at a root point
 n degree of the polynomial
 $'$ first derivative
 $''$ second derivative
 $'''$ third derivative
 $''''$ fourth derivative

APPENDIX B

RATIO OF POLYNOMIAL DERIVATIVE EQUATIONS

Development of General Equations

A general complex number polynomial in terms of its roots can be written as

$$P(z) = (z - b_1)(z - b_2)(z - b_3) \dots (z - b_n) \quad (B1)$$

The first derivative of equation (B1) is

$$P'(z) = \left\{ [(z - b_2)(z - b_3) \dots (z - b_n)] + [(z - b_1)(z - b_3) \dots (z - b_n)] \right. \\ \left. + \dots + [(z - b_1)(z - b_2) \dots (z - b_{n-1})] \right\} \quad (B2)$$

Form the first derivative ratio by dividing equation (B2) by equation (B1)

$$\frac{P'(z)}{P(z)} = f_1 = \frac{1}{z - b_1} + \frac{1}{z - b_2} + \frac{1}{z - b_3} + \dots + \frac{1}{z - b_j} + \dots + \frac{1}{z - b_n} \quad (B3)$$

Generalize equation (B3) to

$$f_1 = \frac{m}{z - b_j} + g_1 \quad (B4)$$

where b_j is the closest root to z , m is the number of roots at b_j , and g_1 is the sum of the remaining $n - m$ terms in equation (B3). Equation (B2) can be written as

$$P'(z) = P(z) \cdot \frac{P'(z)}{P(z)} = P(z) \cdot f_1 \quad (B5)$$

Differentiate equation (B5) to get the second derivative of $P(x)$

$$P''(z) = P'(z) \cdot f_1 + P(z) \cdot f_1' \\ = P(z) \cdot f_1^2 - P(z) \left[\frac{1}{(z - b_1)^2} + \frac{1}{(z - b_2)^2} + \frac{1}{(z - b_3)^2} + \dots + \frac{1}{(z - b_n)^2} \right]$$

$$P''(z) = P(z) \cdot (f_1^2 - f_2) \quad (B6)$$

where

$$f_2 = \left[\frac{1}{(z - b_1)^2} + \frac{1}{(z - b_2)^2} + \frac{1}{(z - b_3)^2} + \dots + \frac{1}{(z - b_n)^2} \right] \quad (B7)$$

Generalize equation (B7) to

$$f_2 = \frac{m}{(z - b_j)^2} + g_2 \quad (B8)$$

The second derivative ratio is

$$\frac{P''(z)}{P'(z)} = \frac{f_1^2 - f_2}{f_1} \quad (B9)$$

Substituting equations (B4) and (B8), equation (B9) becomes

$$\begin{aligned} \frac{P''(z)}{P'(z)} &= \frac{\left(\frac{m}{z - b_j} + g_1 \right)^2 - \left[\frac{m}{(z - b_j)^2} + g_2 \right]}{\frac{m}{z - b_j} + g_1} \\ &= \frac{\frac{m(m-1)}{(z - b_j)^2} + \frac{2mg_1}{z - b_j} + g_1^2 - g_2}{\frac{m}{z - b_j} + g_1} \end{aligned} \quad (B10)$$

Differentiate equation (B6) to get the third derivative of $P(z)$

$$\begin{aligned} P'''(z) &= P'(z) \cdot (f_1^2 - f_2) + P(z) \cdot (2f_1 f_1' - f_2') \\ &= P(z) \cdot f_1 \cdot (f_1^2 - f_2) + P(z) \cdot [2f_1(-1)f_2 - (-2)f_3] \\ &= P(z) \cdot (f_1^3 - 3f_1 f_2 + 2f_3) \end{aligned} \quad (B11)$$

where

$$f_3 = \left[\frac{1}{(z - b_1)^3} + \frac{1}{(z - b_2)^3} + \frac{1}{(z - b_3)^3} + \dots + \frac{1}{(z - b_n)^3} \right] \quad (B12)$$

Generalize equation (B12) to

$$f_3 = \frac{m}{(z - b_j)^3} + g_3 \quad (B13)$$

The third derivative ratio is

$$\frac{P'''(z)}{P''(z)} = \frac{f_1^3 - 3f_1f_2 + 2f_3}{f_1^2 - f_2} \quad (B14)$$

With the generalized f substitutions (B14) becomes

$$\frac{P'''(z)}{P''(z)} = \frac{\left[\frac{m(m-1)(m-2)}{(z-b_j)^3} + \frac{3m(m-1)g_1}{(z-b_j)^2} + \frac{3mg_1^2}{z-b_j} + g_1^3 \right] - 3g_2 \left(\frac{m}{z-b_j} + g_1 \right) + 2g_3}{\frac{m(m-1)}{(z-b_j)^2} + \frac{2mg_1}{z-b_j} + g_1^2 - g_2} \quad (B15)$$

Differentiate equation (B11) to get the fourth derivative of $P(z)$

$$\begin{aligned} P''''(z) &= P'(z) \cdot (f_1^3 - 3f_1f_2 + 2f_3) + P(z) \cdot (3f_1^2f_1' - 3f_2f_1' - 3f_1f_2' + 2f_3') \\ &= P(z) \cdot f_1 \cdot (f_1^3 - 3f_1f_2 + 2f_3) + P(z) \cdot \left[3(f_1^2 - f_2)(-1)f_2 - 3f_1(-2)f_3 + 2(-3)f_4 \right] \\ &= P(z) \cdot (f_1^4 - 6f_1^2f_2 + 8f_1f_3 + 3f_2^2 - 6f_4) \end{aligned} \quad (B16)$$

where

$$f_4 = \left[\frac{1}{(z - b_1)^4} + \frac{1}{(z - b_2)^4} + \frac{1}{(z - b_3)^4} + \dots + \frac{1}{(z - b_n)^4} \right] \quad (B17)$$

Again generalize the f as before

$$f_4 = \frac{m}{(z - b_j)^4} + g_4 \quad (B18)$$

The fourth derivative ratio is

$$\frac{P''''(z)}{P'''(z)} = \frac{f_1^4 - 6f_1^2 f_2 + 8f_1 f_3 + 3f_2^2 - 6f_4}{f_1^3 - 3f_1 f_2 + 2f_3} \quad (B19)$$

With the generalized f substitutions, equation (B19) becomes

$$\begin{aligned} \frac{P''''(z)}{P'''(z)} = & \frac{\left[\frac{m(m-1)(m-2)(m-3)}{(z-b_j)^4} + \frac{4m(m-1)(m-2)g_1}{(z-b_j)^3} + \frac{6m(m-1)g_1^2}{(z-b_j)^2} + \frac{4mg_1^3}{z-b_j} + g_1^4 \right]}{\left[\frac{m(m-1)(m-2)}{(z-b_j)^3} + 3 \frac{m(m-1)g_1}{(z-b_j)^2} + 3 \frac{mg_1^2}{z-b_j} + g_1^3 \right] - 3g_2 \left(\frac{m}{z-b_j} + g_1 \right) + 2g_3} \\ & + \frac{-6g_2 \left[\frac{m(m-1)}{(z-b_j)^2} + \frac{2mg_1}{z-b_j} + g_1^2 \right] + 8g_3 \left(\frac{m}{z-b_j} + g_1 \right) + 3g_2^2 - 6g_4}{\left[\frac{m(m-1)(m-2)}{(z-b_j)^3} + 3 \frac{m(m-1)g_1}{(z-b_j)^2} + 3 \frac{mg_1^2}{z-b_j} + g_1^3 \right] - 3g_2 \left(\frac{m}{z-b_j} + g_1 \right) + 2g_3} \end{aligned} \quad (B20)$$

Higher derivatives develop in the same way. The process follows the pattern used to get the first four derivatives. The higher derivatives have progressively more terms in the generalized forms for the f 's. This does not seem consistent with the fact that $P^n(z)$ must be a constant and all the higher derivatives must be zero. But what is happening is that in progressing to high derivatives with the generalized forms for the f 's, more and more terms are carried internally that would cancel if the f 's and g 's were expanded in their $z - b$ terms.

At this point a generalization of the patterns shown by the first four derivatives is in order. First note that the terms in the brackets in equation (B20) and in each of the corresponding lower order equations correspond to something like a binomial expansion. The general form for f_1 to the k^{th} power can be expressed as

$$f_1^k = \sum_{l=0}^k \frac{k! m! (g_1)^l}{l! (k-l)! (m-k+l)! (z-b_j)^{k-l}} \quad (B21)$$

if $0!$ is understood to be one.

Comparison of equations (B19) and (B20) and each of the other corresponding pairs of derivative ratio equations indicates the following procedure for transferring the derivative ratio equations from those with f 's to the form with m 's and g 's: (1) replace the f_1 's and powers of f_1 with equation (B21), and (2) replace all the f 's that do not have a subscript of 1 with g 's that have the corresponding subscript.

Division of a Real Root from the Polynomials (from ref. 3)

The general polynomial

$$P(x) = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1} \quad (B22)$$

can be expressed in terms of a root as

$$P(x) = (x - x_r) (c_1 x^{n-1} + c_2 x^{n-2} + \dots + c_{n-1} x + c_n) + c_{n+1} \quad (B23)$$

By equating powers of x in equations (B22) and (B23)

$$\left. \begin{aligned} a_1 &= c_1 \\ a_2 &= c_2 - c_1 x_r \\ a_3 &= c_3 - c_2 x_r \\ &\vdots \\ a_n &= c_n - c_{n-1} x_r \\ a_{n+1} &= c_{n+1} - c_n x_r \end{aligned} \right\} \quad (B24)$$

The c 's can be solved for directly by starting from the top of the equation set (B24)

$$c_1 = a_1$$

$$c_2 = a_2 + c_1 x_r$$

$$c_3 = a_3 + c_2 x_r$$

.

.

.

$$c_{n+1} = a_{n+1} + c_n x_r$$

The constants, c_1 to c_n , are the coefficients for the polynomial left after the root is divided out. The constant c_{n+1} is the remainder which should be zero.

Complex Number Root Coordinate Adjustment by Ratio of Derivatives

When in the vicinity of roots, complex derivative ratios are used for the analysis and finer adjustment to root coordinates. Once m is determined, the coordinate adjustment equations are obtained from the following general equations:

$$\frac{P^m(z)}{P^{m-1}(z)} \approx \frac{1}{z - z_{\text{new}}} \quad (\text{B26})$$

Break equation (B26) into real and imaginary parts

$$\frac{[P^m(x) + iP^m(y)]}{[P^{m-1}(x) + iP^{m-1}(y)]} \frac{[P^{m-1}(x) - iP^{m-1}(y)]}{[P^{m-1}(x) - iP^{m-1}(y)]} = \frac{1}{[(x - x_{\text{new}}) + i(y - y_{\text{new}})]} \frac{[(x - x_{\text{new}}) - i(y - y_{\text{new}})]}{[(x - x_{\text{new}}) - i(y - y_{\text{new}})]}$$

$$\frac{P^m(x)P^{m-1}(x) + P^m(y)P^{m-1}(y) + i[P^m(y)P^{m-1}(x) - P^m(x)P^{m-1}(y)]}{[P^{m-1}(x)]^2 + [P^{m-1}(y)]^2} = \frac{(x - x_{\text{new}}) - i(y - y_{\text{new}})}{(x - x_{\text{new}})^2 + (y - y_{\text{new}})^2}$$

Let

$$A = \frac{P^m(x)P^{m-1}(x) + P^m(y)P^{m-1}(y)}{[P^{m-1}(x)]^2 + [P^{m-1}(y)]^2} \quad (\text{B27})$$

$$B = \frac{P^m(y)P^{m-1}(x) - P^m(x)P^{m-1}(y)}{[P^{m-1}(x)]^2 + [P^{m-1}(y)]^2} \quad (B28)$$

Therefore,

$$A + iB = \frac{(x - x_{\text{new}}) - i(y - y_{\text{new}})}{(x - x_{\text{new}})^2 + (y - y_{\text{new}})^2}$$

$$A = \frac{x - x_{\text{new}}}{(x - x_{\text{new}})^2 + (y - y_{\text{new}})^2} \quad (B29)$$

$$B = - \frac{y - y_{\text{new}}}{(x - x_{\text{new}})^2 + (y - y_{\text{new}})^2} \quad (B30)$$

By dividing equation (B29) by equation (B30), we have the following simple relation:

$$\frac{A}{B} = - \frac{x - x_{\text{new}}}{y - y_{\text{new}}} \quad (B31)$$

Equation (B31) then can be used in equations (B29) and (B30) to get the following x and y adjustment equations:

$$x_{\text{new}} = x - \frac{A}{A^2 + B^2} \quad (B32)$$

$$y_{\text{new}} = y + \frac{B}{A^2 + B^2} \quad (B33)$$

Estimate of Next Nearest Complex Root

When near a root, the term g_1 is obtained from

$$g_1 = \frac{1}{m+1} \frac{P^{m+1}(z)}{P^m(z)}$$

where

$$g_1 = \sum_{j=1}^{n-m} \frac{1}{z - b_j}$$

Included in the summation is the conjugate of the present root. If the next nearest root and its conjugate are relatively much nearer than the others remaining, g_1 can be approximated by

$$g_1 = g_R + ig_I = \frac{m}{z_r - \bar{z}_r} + \frac{1}{z_r - z_{next}} + \frac{1}{z_r - \bar{z}_{next}}$$

where g_R and g_I are, respectively, the real and imaginary parts of g_1 . Now continue to expand the preceding equation as follows;

$$\begin{aligned} g_R + ig_I &= \frac{m}{(x_r + iy_r) - (x_r - iy_r)} + \frac{1}{(x_r + iy_r) - (x_{next} - iy_{next})} + \frac{1}{(x_r + iy_r) - (x_{next} - iy_{next})} \\ &= \frac{m}{2iy_r} + \frac{1}{(x_r - x_{next}) + i(y_r - y_{next})} + \frac{1}{(x_r - x_{next}) + i(y_r + y_{next})} \\ &= \frac{-im}{2y_r} + \frac{[(x_r - x_{next}) + i(y_r + y_{next})] + [(x_r - x_{next}) + i(y_r - y_{next})]}{[(x_r - x_{next}) + i(y_r - y_{next})][(x_r - x_{next}) + i(y_r + y_{next})]} \\ &= \frac{-im}{2y_r} + \frac{2(x_r - x_{next}) + 2iy_r}{(x_r - x_{next})^2 - (y_r^2 - y_{next}^2) + 2i(x_r - x_{next})y_r} \\ &= \frac{-im}{2y_r} + \frac{2[(x_r - x_{next}) + iy] [(x_r - x_{next})^2 - (y_r^2 - y_{next}^2) - 2i(x_r - x_{next})y_r]}{[(x_r - x_{next})^2 - (y_r^2 - y_{next}^2)]^2 + 4(x_r - x_{next})^2 y_r^2} \\ &= \frac{-im}{2y_r} + \frac{2\{(x_r - x_{next})[(x_r - x_{next})^2 - (y_r^2 - y_{next}^2)] + 2(x_r - x_{next})y_r^2\}}{D} \\ &\quad + \frac{2iy_r\{[(x_r - x_{next})^2 - (y_r^2 - y_{next}^2)] - 2(x_r - x_{next})^2\}}{D} \end{aligned}$$

where

$$D = \left[(x_r - x_{\text{next}})^2 - (y_r^2 - y_{\text{next}}^2) \right]^2 + 4(x_r - x_{\text{next}})^2 y_r^2$$

The separate equations for real and imaginary parts are

$$g_R = \frac{2(x_r - x_{\text{next}}) \left[(x_r - x_{\text{next}})^2 + (y_r^2 + y_{\text{next}}^2) \right]}{D} \quad (\text{B34})$$

$$g_I + \frac{m}{2y_r} = - \frac{2y_r \left[(x_r - x_{\text{next}})^2 + (y_r^2 - y_{\text{next}}^2) \right]}{D} \quad (\text{B35})$$

Divide the preceding equations and solve for y_{next}^2

$$\frac{g_R}{g_I + \frac{m}{2y_r}} = - \frac{(x_r - x_{\text{next}}) \left[(x_r - x_{\text{next}})^2 + (y_r^2 + y_{\text{next}}^2) \right]}{y_r \left[(x_r - x_{\text{next}})^2 + (y_r^2 - y_{\text{next}}^2) \right]}$$

$$-g_R y_r \left[(x_r - x_{\text{next}})^2 + (y_r^2 - y_{\text{next}}^2) \right] = \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}}) \left[(x_r - x_{\text{next}})^2 + y_r^2 + y_{\text{next}}^2 \right]$$

$$y_{\text{next}}^2 \left[g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}}) \right] = \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}}) \times \left[(x_r - x_{\text{next}})^2 + y_r^2 \right] + g_R y_r \left[(x_r - x_{\text{next}})^2 + y_r^2 \right]$$

$$y_{\text{next}}^2 = \frac{\left[g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}}) \right] \left[(x_r - x_{\text{next}})^2 + y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{\text{next}})} \quad (\text{B36})$$

Substitute equation (B36) into equation (B34) and solve for x_{next}

$$\begin{aligned}
g_R &= \frac{2(x_r - x_{next}) \left\{ (x_r - x_{next})^2 + y_r^2 + \frac{\left[g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right] \left[(x_r - x_{next})^2 + y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next})} \right\}}{\left\{ (x_r - x_{next})^2 - y_r^2 + \frac{\left[g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right] \left[(x_r - x_{next})^2 + y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next})} \right\}^2 + 4(x_r - x_{next})^2 y_r^2} \\
&= \frac{2(x_r - x_{next}) \left\{ \frac{2g_R y_r \left[(x_r - x_{next})^2 + y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next})} \right\}}{\left\{ \frac{2 \left[(x_r - x_{next})^2 g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) y_r^2 \right]}{g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next})} \right\}^2 + 4(x_r - x_{next})^2 y_r^2} \\
&= \frac{2(x_r - x_{next}) (2g_R y_r) \left[(x_r - x_{next})^2 + y_r^2 \right] \left[g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right]}{\left\{ 2 \left[(x_r - x_{next})^2 g_R y_r + \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) y_r^2 \right] \right\}^2 + 4(x_r - x_{next})^2 y_r^2 \left[g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right]^2} \\
&= \frac{g_R \left[g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right]}{y_r (x_r - x_{next}) \left[g_R^2 + \left(g_I + \frac{m}{2y_r} \right)^2 \right]}
\end{aligned}$$

$$g_R y_r (x_r - x_{next}) \left[g_R^2 + \left(g_I + \frac{m}{2y_r} \right)^2 \right] = g_R \left[g_R y_r - \left(g_I + \frac{m}{2y_r} \right) (x_r - x_{next}) \right]$$

$$(x_r - x_{next}) \left\{ y_r \left[g_R^2 + \left(g_I + \frac{m}{2y_r} \right)^2 \right] + g_I + \frac{m}{2y_r} \right\} = g_R y_r$$

$$x_{\text{next}} = x_r - \frac{g_R y_r}{y_r \left[g_R^2 + \left(g_I + \frac{m}{2y_r} \right)^2 \right] + g_I + \frac{m}{2y_r}} \quad (\text{B37})$$

The plus y_{next} value is used in equation (B36) in the attempt to keep the search for complex roots above the x-axis.

Division of a Complex Root from the Polynomial

The general polynomial

$$P(z) = a_1 z^n + a_2 z^{n-1} + \dots + a_n z + a_{n+1} \quad (\text{B38})$$

can be expressed in terms of a root as

$$P(z) = (z - z_r) (c_1 z^{n-1} + c_2 z^{n-2} + \dots + c_{n-1} z + c_n) + c_{n+1}$$

Equating powers of z results in the following:

$$\left. \begin{aligned} a_1 &= c_1 \\ a_2 &= c_2 - c_1 z_r \\ a_3 &= c_3 - c_2 z_r \\ &\vdots \\ a_n &= c_n - c_{n-1} z_r \\ a_{n+1} &= c_{n+1} - c_n z_r \end{aligned} \right\} \quad (\text{B39})$$

The c 's are complex coefficients which may be solved for directly by starting from the top of the equation set (B39).

$$c_1 = a_1$$

$$c_2 = a_2 + c_1 z_r$$

$$c_3 = a_3 + c_2 z_r$$

.

.

.

$$c_{n+1} = a_{n+1} + c_n z_r$$

so

$$c_{R, 1} = a_1$$

$$c_{R, 2} = a_2 + c_{R, 1} x_r - c_{I, 1} y_r$$

$$c_{R, 3} = a_3 + c_{R, 2} x_r - c_{I, 2} y_r$$

.

.

.

$$c_{R, n+1} = a_{n+1} + c_{R, n} x_r - c_{I, n} y_r$$

and

$$c_{I, 1} = 0$$

$$c_{I, 2} = c_{R, 1} y_r + c_{I, 1} x_r$$

$$c_{I, 3} = c_{R, 2} y_r + c_{I, 2} x_r$$

.

.

.

$$c_{I, n+1} = c_{R, n} y_r + c_{I, n} x_r$$

The sets of constants $c_{R, 1}$ to $c_{R, n}$ and $c_{I, 1}$ to $c_{I, n}$, are the respective real and imaginary sets of coefficients for the polynomial left after the root is divided out. This division is used only to determine the remainder terms $c_{R, n+1}$ and $c_{I, n+1}$.

Division of a Complex Conjugate Pair of Roots from the Polynomial

The general polynomial (eq. (B38)) can be expressed in terms of a root and its complement as

$$P(z) = (z - z_r)(z - \bar{z}_r) (c_1 z^{n-2} + c_2 z^{n-3} + \dots + c_{n-2} z + c_{n-1}) + c_n z + c_{n+1}$$

The product, $(z - z_r)(z - \bar{z}_r)$, can be expressed as

$$\begin{aligned} (z - \bar{z}_r)(z - z_r) &= z^2 - z(z_r + \bar{z}_r) + z_r \bar{z}_r \\ &= (x + iy)^2 - (x + iy)2x_r + (x_r + iy_r)(x_r - iy_r) \\ &= x^2 - y^2 - 2xx_r + x_r^2 + y_r^2 + 2iy(x - x_r) \end{aligned} \quad (B41)$$

Since the problem being solved is by definition a real polynomial with x the only independent variable, a general y does not exist. Therefore, equation (B41) can be written in real algebra.

$$(z - z_r)(z - \bar{z}_r) = x^2 - 2xx_r + x_r^2 + y_r^2$$

The general polynomial can be written as

$$P(x) = (x^2 - 2xx_r + x_r^2 + y_r^2) (c_1 x^{n-2} + c_2 x^{n-3} + \dots + c_{n-1}) + c_n x + c_{n+1}$$

Equating powers of x

$$\begin{aligned}
a_1 &= c_1 \\
a_2 &= c_2 - 2c_1x_r \\
a_3 &= c_3 - 2c_2x_r + c_1(x_r^2 + y_r^2) \\
a_4 &= c_4 - 2c_3x_r + c_2(x_r^2 + y_r^2) \\
&\cdot \\
&\cdot \\
&\cdot \\
a_n &= c_n - 2c_{n-1}x_r + c_{n-2}(x_r^2 + y_r^2) \\
a_{n+1} &= c_{n+1} + c_{n-1}(x_r^2 + y_r^2)
\end{aligned} \tag{B42}$$

The equation set (B42) can be solved directly for the c 's by starting from the top.

$$\begin{aligned}
c_1 &= a_1 \\
c_2 &= a_2 + 2c_1x_r \\
c_3 &= a_3 + 2c_2x_r - c_1(x_r^2 + y_r^2) \\
c_4 &= a_4 + 2c_3x_r - c_2(x_r^2 + y_r^2) \\
&\cdot \\
&\cdot \\
&\cdot \\
c_n &= a_n + 2c_{n-1}x_r - c_{n-2}(x_r^2 + y_r^2) \\
c_{n+1} &= a_{n+1} - c_{n-1}(x_r^2 + y_r^2)
\end{aligned} \tag{B43}$$

In this case the polynomial remainder after the root and its complement are divided out is

$$c_n x_r + c_{n+1} \tag{B44}$$

APPENDIX C

THE PROGRAM

Information for Users

The computer program is written in FORTRAN IV language and it is run in double precision. On an IBM direct coupled 7044 - 7094 system the running time averages about 0.01 minute for an eighth-degree polynomial. The program has one systems subroutine called DUBIO which appears before the input READ statement. The function of this subroutine is to allow double precision input and output formats on the NASA Lewis computer. It is not needed on most other computers. For some applications it may not be necessary to use the double precision input and output, but one should always remember that the output accuracy is a function of input accuracy.

The first card of an input data set has the number N of polynomial coefficients expected on the following data cards. The polynomial coefficients A(I) are read in order beginning with the high degree end of the polynomial. The input format is shown in figure 5.

TITLE		PROJECT NUMBER		ANALYST		SHEET OF	
Input data							
STATEMENT NUMBER	8	FORTRAN STATEMENT					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	IDENTIFICATION						
Field for input parameters							
N							First card
	A(1)	A(1)				A(3)	Second card
	A(N-1)	A(N)					Last card
Data format							
I3	D24.16	D24.16				D24.16	
	D24.16	D24.16					
Example: $P(z) = (x + 1)(x - 1)(x + 0.000001 + 1)(x + 0.000001 - 1)$							
5							
0.1		D+01	0.2		D-05	0.20000000000001	D+01
0.2		D-05	0.1		D+01		

NASA-C-856 (REV. 9-16-59)

Figure 5. - Input data.

The polynomial relative error criterion constants, TOLRQ and TOLIQ, which are defined near the beginning of the main program ROOTS, are based on 16-significant-figure double precision capability. If the program is to be used with other than 16-significant-figure machine capability, the constants in TOLRQ and TOLIQ should be adjusted directly with the change in significant figure capability.

Note that the way the program is structured, meaningful analysis can only be done if TOLRQ and TOLIQ are above the absolute error levels of computed polynomial values. However, raising TOLRQ and TOLIQ decreases the root resolution capability of a program. To allow for near maximum program root resolution capability, TOLRQ and TOLIQ are internally adjusted to account for the greater number of computations needed to evaluate a higher degree polynomial. In the Polynomial and Polynomial Derivative Evaluation section's discussion of ROOTS it was pointed out that the least capability occurs when roots are grouped. The TOLRQ and TOLIQ adjustments with polynomial degree are based on roots being grouped up to degree ten. For higher degree polynomials the probability of groups of more than ten roots is small, so the increase of TOLRQ and TOLIQ with polynomial degree were somewhat leveled off.

A final comment concerns the use of ARATIO contour plots if a user is not satisfied with the computed roots or has reason to believe that a ratio of derivatives program is not working properly for a particular problem. The polynomial coefficients can be used in a little program to compute ARATIO values for two or three ND values on a x - y grid over the troublesome area. Contour plots from the grid values can be a very helpful aid in indicating where the roots should be. Values of POLY(ND - 1) in the absolute error range will fog the issue, but such ranges are usually obvious on a plot.

Description of Program Variables

Symbol	Description
A(I)	polynomial coefficients beginning from high degree end
ARATIO	absolute value of the ratio of successive ND derivative ratios, $\frac{[POLY(ND + 1)]/[POLY(ND)]}{[POLY(ND)]/[POLY(ND - 1)]}$
B(I)	coefficients of polynomial derivative
CA(I)	polynomial normalizing constant; also the calculated imaginary part of polynomial coefficients at a complex root
CRATIO(I)	imaginary part of a complex number derivative ratio
D	value of the complex polynomial, POLY(ND - 1), squared

DELTX	real component of an independent variable step in the complex plane
DELT	imaginary component of an independent variable step in the complex plane
DIR	step direction indicator in the Taylor's series search for real roots
DRATIO	ratio of successive ND derivative ratios, $\frac{[POLY(ND + 1)/POLY(ND)]}{[POLY(ND)/POLY(ND - 1)]}$
DTHETA	spiral angular increment (30^0) used in moving toward a root when known to be between nearby roots in the complex plane
DX	x-increment to estimated location of next root; also, a reference x step when searching for a root known to be within a root group
DXT	normalized real part of complex number polynomial first derivative as used in complex number Taylor's series solution for a new z
DYT	normalized imaginary part of complex number polynomial first derivative as used in complex number Taylor's series solution for a new z
DZT	relative distance to root with plus sign in complex number Taylor's series equation
DZT2	relative distance to root with minus sign in complex number Taylor's series equation
D1	value of complex polynomial, POLY(ND), squared
D2	value of the ND complex derivative ratio squared
D3	value of the ND - 1 complex derivative ratio squared
FACT	ratio of x adjustment to previous x adjustment in Taylor's series approach to a root
FK	multiple of coefficient for calculation of polynomial derivative
H	x adjustment toward a root by the Taylor's series approximation
HX	x adjustment toward a root by complex Taylor's series approximation
HY	y adjustment toward a root by complex Taylor's series approximation
HYP	magnitude of the square root term in complex plane Taylor's series approximation for a new z
H1	x backsteps in Taylor's series search for real roots
I	polynomial coefficient subscript for dimensional variables
ICHECK	a logical parameter which activates a running record and update of XHIGH and XLOW

IDIR	logical direction indicator in the Taylor's series search for real roots
II	index of previous nonzero polynomial coefficient in the computation of a polynomial value
IMSURE	logical indicator of the certainty of a real root remaining in the polynomial
INK	logical indicator of the direction of scaling
IR	system input tape number
IREAL	logical indicator of the type of root sought (IREAL = 1 for real roots, IREAL = 0 for complex roots)
ISIGN	routing device for setting spiral direction
ITRIG	routing device of how polynomial approaches the x-axis in the Taylor's series search for real roots
IW	system output tape number
IXM	logical indicator used when a pair of root candidates need to be investigated
J	current number of roots found
JTRIG	routing device
K	index used in dividing out roots
KTRIG	routing device in ratio of derivative analysis for roots that are difficult to resolve
L	logical device for counting the number of successive zero coefficients in a polynomial
LIMIT	logical indicator that a computed polynomial value is probably in the absolute error range
LTRIG	logical device and counter used when known to be between or among roots
M	degree of polynomial derivative
MULT	number of roots at a root point
N	number of polynomial coefficients (degree of the polynomial plus one)
NC	index used in scaling a polynomial, also a counter during preliminary checks
ND	highest order polynomial derivative needed in current analysis
NDEG	degree of the current polynomial
NDRV	order of the polynomial derivative

NEXP	exponent hexadecimal order of magnitude factor used in polynomial scaling
NEXPS	exponent hexadecimal order of magnitude factor used in initial polynomial scaling
NK	index used in polynomial scaling
NM	index used in scaling a polynomial, also a counter during preliminary checks
NND	temporary storages of ND
NUM	number of roots in the polynomial
OARAT	value of ARATIO for previous iteration
OAX	reference value of ARATIO for the x-step in setting up the spiral reference angle
OAY	value of ARATIO from the x-step in setting up the spiral reference angle
ODRAT	value of DRATIO for previous iteration
OPOLY1	most recent good value of POLY(1)
OPOLY2	most recent good value of POLY(2)
ORATIO	value of RATIO(ND - 1) from previous iteration
PHI	complex plane angle of square root term in complex number Taylor's series solution for a new z
P(I)	normalized polynomial, P(1), or a polynomial derivative when $I > 1$
POLY(I)	polynomial, POLY(1), or a polynomial derivative when $I > 1$
POLY1R	ratio of present polynomial value to last Taylor's series iteration polynomial value
PXHIGH	value of normalized polynomial at XHIGH
PXLOW	value of normalized polynomial at XLOW
Q	square root term of quadratic equation
Q1	square root term of Taylor's series approximation for real root location
RA	polynomial ratio used to estimate location of next root
RA(I)	temporarily real part of polynomial coefficient at a complex root
RAD	radial component of a spiral in the complex plane
RATIO(I)	ratio of a polynomial derivative to the next lower order polynomial derivative

RATIOI	imaginary part of polynomial ratio used to estimate location of the next root
RATIOR	real part of polynomial ratio used to estimate location of next root
RRATIO(I)	real part of complex number derivative ratio
SCALE	hexadecimal order of magnitude factor used to scale polynomial roots
SLOPE	slope of least squares line fit of the hexadecimal logarithms of the polynomial coefficients
SRATIO	parameter used in setting step size when between or among roots of a group, $\sqrt{\text{POLY}(\text{ND} + 1) * \text{POLY}(\text{ND} - 1)}$
SXI	sum of XN for determination of SLOPE
SXIYI	sum of the product of XN times YI
SXI2	SXI squared
SYI	sum of YI for determination of SLOPE
THETA	local angular coordinate of a spiral in the complex plane
THETAO	reference angle of a spiral in the complex plane
TOLI	complex root resolution distance criterion
TOLIQ	probable absolute error of a computed normalized polynomial value in the complex plane
TOLR	real root resolution distance criterion
TOLRQ	probable absolute error of a computed normalized polynomial value along the x-axis
X	independent polynomial variable or real component of independent complex variable
XCON	one of the constants used to determine DELTX and DELTY for a spiral
XCPLX	x-coordinate at which the search for complex roots is begun
XDT	real part of the denominator of the complex Taylor's series approximation for a new z
XH	temporary new value of x
XHIGH	closest x definitely known to be greater than the value of the root sought
XI	initial x value in the search for a new root
XL	reference x when adjusting to a new x value between XHIGH and XLOW
XLAST	x value of a previous iteration

XLOW	closest x definitely known to be less than the value of the root sought
XMULT	number of roots at a root point
XN	number of a polynomial coefficient, which is the independent variable in the determination of SLOPE
XNT	real part of the numerator of the complex Taylor's series approximation for a new z
XOLD	x value of a previous iteration during resolution of a root
XP(I)	real part of normalized complex polynomial or polynomial derivative
XPN	temporary storage of x or XP(I)
XPOLE	x -coordinate of spiral pole in the complex plane
XPOLY(I)	real part of complex polynomial or polynomial derivative
XQ	real part of square root term in the complex number Taylor's series solution for a new z
XR(I)	polynomial remainder or real part of polynomial remainder after a root is divided out
XROOT(I)	x -coordinate of a root
XX	temporary storage of x
Y	imaginary component of independent complex variable
YCON	one of the constants used to determine DELTX and DELTY for a spiral
YCPLX	y -coordinate at which the search for complex roots is begun
YDT	imaginary part of denominator of the complex number Taylor's series approximation for a new z
YI	initial y value in the search for a new root, also hexadecimal logarithm of a polynomial coefficient
YLAST	y value of previous iteration on the spiral
YNT	imaginary part of the numerator of the complex Taylor's series approximation for a new z
YOLD	y value of a previous iteration during resolution of a root
YP(I)	imaginary part of normalized complex polynomial or polynomial derivative
YPOLE	y -coordinate of spiral pole in the complex plane
YPOLY(I)	imaginary part of complex polynomial or polynomial derivative

YQ imaginary part of square root term in the Taylor's series approximation
 for a new z

YR(I) imaginary part of the polynomial remainder after a root is divided out

YROOT(I) y-coordinate of a root

YY temporary storage of y

ZCON one of the constants used to set up the spiral increment

Program Listing

\$IBFIC RCCTS

```

      DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
1  RA, RATIO, RRATIO, SCALE, X, XHIGH, XLAST, XLOW, XP, XPN, XPOLY,
2  XR, XRCCT, Y, YP, YPOLY, YR, YROOT
      COMMON /POLYN/ ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
1  ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
2  NDEG, NERV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,
3  TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
4  A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
5  RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
6  YP(100), YPOLY(100), YR(100), YROOT(100)
      ITEST = 1
      IF (ITEST.EQ.2) CALL DUBIO

C ---      A(I) ARE COEFFICIENTS FOR THE POLYNOMIAL, POLY(X) = A(N)*X**N
C --- + A(N-1)*X**(N-1) + ---- + A(2)*X**2 + A(1)*X + A(0). THE COEFFI-
C --- CIENTS MUST BE INPUT IN ORDER. START THEM FROM THE HIGH DEGREE
C --- END OF THE POLYNOMIAL.

      IR = 5
      IW = 6
5  READ (IR,1000) N, (A(I),I=1,N)
      IF (N.LE.1) WRITE (IW,1010)
      IF (N.GT.100) WRITE (IW,1020)
      WRITE (IW,1030) (A(I),I=1,N)
      NUM = N - 1
      J = 0
      NC = NUM
      IF (NC.GT.10) NC = 9 + NUM/10
      TOLRQ = 5.0*2.0**NC*1.0E-16
      TOLR = SQRT(TOLRQ)
      TCLIQ = 2.0**NC*1.0E-15
      TCLI = SQRT(TOLIQ)
      NC = 1

C ---      IF THE POLYNOMIAL BEGINS WITH ZERO COEFFICIENTS LOWER THE
C --- DEGREE APPROPRIATELY.

```

```

      DC 6 I=1,N
      IF (A(I).NE.0.0) GO TO 7
      NC = NC + 1
6     CCNTINUE
      GC TC 5
7     IF (NC.EQ.1) GO TO 9
      DC 8 I=NC,N
      NM = I - NC + 1
8     A(NM) = A(I)
      N = N - NC + 1
      NUM = N - 1
9     XI = 0.0
      NDEG = N-1
C ---      TAKE CUT ANY ZERO ROOTS FIRST.

      FK = ABS(A(1))
      DC 10 I=2,NDEG
      IF (ABS(A(I)).LT.FK*0.01*TOLRQ) GO TO 10
      FK = ABS(A(I))
10    CCNTINUE
      IF (ABS(A(N)/FK).GT.0.01*TOLRQ) GO TO 12
      X = 0.0
      J = J + 1
      XRCCT(J) = 0.0
      YRCCT(J) = 0.0
      XR(J) = 0.0
      YR(J) = 0.0
      N = N - 1
      GC TC 9
12    IREAL = 1

C ---      SCALE POLYNOMIAL SO ROOTS ARE APPROXIMATELY OF ORDER ONE FOR
C ---      RCCT RESOLUTION COMPUTATION PURPOSES. A LEAST SQUARES FIT OF THE
C ---      LCGS OF THE POLYNOMIAL COEFFICIENTS TO A LINE IS USED.

13    SXI = 0.0
      SXI2 = 0.0
      SYI = 0.0
      SXIYI = 0.0
      NC = 0
      FK = ABS(A(1))
      DC 14 I=1,N
      IF (ABS(A(I)).LT.FK*0.01*TOLRQ) GO TO 14
      NC = NC + 1
      FK = ABS(A(I))
      XN = FLOAT(I)
      YI = 0.36067376*ALOG(FK)
      SXI = SXI + XN
      SXI2 = SXI2 + XN**2
      SYI = SYI + YI
      SXIYI = SXIYI + XN*YI
14    CCNTINUE
      SLCPE = (SYI/FLOAT(NC) - SXIYI/SXI)/(SXI/FLOAT(NC) - SXI2/SXI)
      FK = 0.36067376*ALOG(ABS(A(1)))
      NC = FK
      NK = 0
      INK = 1
      IF (SLOPE.GT.0.0) GO TO 15
      SLCPE = -SLOPE
      INK = -1

```

```

15 DC 16 I=1,10
   IF (SLOPE.LT.0.5) GO TO 17
   NK = NK + INK
16 SLCPE = SLCPE - 1.0
17 NEXP = NK
   NEXPS = NEXP
   SCALE = 16.000**NEXP
   IF (NK.EQ.0.AND.NC.EQ.0) GO TO 19
   DC 18 I=1,N
   NM = NK*(I-1) + NC
18 A(I) = A(I)/16.000**NM
19 IF (N.LE.3) GO TO 570
   IF (IREAL.EQ.0) GO TO 280
   FACT = 2.0
   IDIR = 1
   XCPLX = 0.0
   YCPLX = 1.0

```

```

C ---      INITIALIZE PARAMETERS.  X IS THE INDEPENDENT VARIABLE.  J IS
C --- THE NUMBER OF ROOTS FOUND.  FACT IS AN X STEP SIZE FACTOR.  IDIR
C --- INDICATES THE DIRECTION TO MOVE ALONG X.  IREAL IS A TRIGGER TO
C --- INDICATE THE SEARCH FOR REAL OR IMAGINARY ROOTS.  ONLY REAL
C --- ALGEBRA IS USED WHEN IREAL IS A POSITIVE INTEGER.  H IS A DELTA X
C --- INCREMENT.  ITRIG IS A ROUTING DEVICE.  IT BASICALLY GIVES AN
C --- INDICATION OF HOW THE POLYNOMIAL APPROACHES THE X AXIS.  JTRIG IS
C --- A ROUTING DEVICE WHEN HIGHER DERIVATIVES ARE NEEDED.

```

```

20 H = 0.0
   DIR = IDIR
   IMSURE = 0
   ITRIG = 0
   IXM = 0
   JTRIG = 0
   KTRIG = 0
   LTRIG = 0
   X = XI

```

```

C ---      ND IS THE NUMBER OF DERIVATIVES CALCULATED.
C ---      NDRV IS A PARTICULAR DERIVATIVE.

```

```

25 ND = 2

```

```

C ---      THIS PROGRAM USES A QUOTIENT OF SUCCESSIVE DERIVATIVES METHOD
C --- IN BOTH THE REAL AND COMPLEX REGIMES TO PINPOINT THE VALUES AND
C --- THE MULTIPLICITY OF THE ROOTS.  SINCE THE REAL ROOTS ARE EASIER TO
C --- WORK WITH, THEY ARE FOUND FIRST.  IF THERE ARE REAL ROOTS THE
C --- DEGREE OF THE POLYNOMIAL IS DECREASED BEFORE COMPLEX NUMBERS ARE
C --- USED.

```

```

28 ICHECK = 0
30 DC 32 I=1,N
32 B(I) = A(I)
   NDEG = N-1
   M = NDEG
   NDRV = 0
35 CALL RPCLY
   IF (ICHECK.EQ.0.OR.ABS(P(1)).GT.0.5*TOLRQ) GO TO 40
   IF (ICHECK.GT.0) GO TO 38

```

```

36 X = (XLOW + X)/2.0
   GO TO 28
38 X = (XHIGH + X)/2.0
   GO TO 28
40 ICHECK = 0
   IF (ITRIG.EQ.0.OR.ABS(P(1)).LE.0.5*TOLRQ) GO TO 48
   IF (IMSURE.EQ.0) GO TO 44
   IF (P(1)/PXHIGH.LT.0.0) GO TO 42
   XHIGH = X
   PXHIGH = P(1)
   GO TO 48
42 XLOW = X
   PXLW = P(1)
   GO TO 48
44 IF (PCLY(1)/OPOLY1.GT.0.0) GO TO 48
   IMSURE = 1
   IF (DELTX.LT.0.0) GO TO 46
   XHIGH = X
   PXHIGH = P(1)
   XLOW = XLAST
   PXLW = (PCLY1/B(N))
   GO TO 48
46 XHIGH = XLAST
   PXHIGH = OPOLY1/B(N)
   XLOW = X
   PXLW = P(1)
48 IF (ABS(P(1)).LT.TOLR.OR.JTRIG.EQ.2) GO TO 160
   IF (IREAL.EQ.0) GO TO 50
   IF (ITRIG.NE.0.OR.ABS(P(2)).GT.TOLR) GO TO 50
   H = 0.1*DIR
   IF (ABS(POLY(1)).GE.ABS(POLY(3))) GO TO 49
   IF (POLY(1)/POLY(3).LT.0.0) H = -DIR/(POLY(1)*POLY(3))
49 DELTX = 1
   DIR = 1.C
   ITRIG = -1
   GO TO 102

```

C --- REAL ROOTS ARE THE X VALUES WHICH MAKE P(1) = 0.0 START AT
C --- X = 0.0. THE NEXT X IS DETERMINED WITH A 3 TERM TAYLORS SERIES.

```

50 Q1 = (POLY(2)/POLY(3))**2 - 2.0*POLY(1)/POLY(3)

```

C --- IF Q1 IS LESS THAN ZERO, P(2) MAY CHANGE SIGN BEFORE P(1)
C --- CROSSES THE AXIS.

```

   IF (Q1.LT.0.0.OR.ITRIG.EQ.1) GO TO 70
   IF (JTRIG.EQ.0) GO TO 57
   IF (PCLY(1)/OPOLY1.LT.0.0) GO TO 56
   IF (ITRIG.LT.0) GO TO 52
   IF (POLY(2)/POLY(1)*H) 57,57,66
52 IF (POLY(2)/OPOLY2.GE.0.0) GO TO 57
56 DIR = 1.C
57 ITRIG = -1.0
58 IF (DABS(P(2)).LT.TOLRQ) POLY(2) = TOLRQ*CA(2)
   IF (ABS(POLY(3)/POLY(2)).GT.0.01) GO TO 60
59 H = -POLY(1)/POLY(2)*DIR
   IF (X) 64,65,64

```



```

60 IF (ABS(F(3)).LT.TOLRQ) GO TO 62
   IF (ABS(F(2)).LT.TOLRQ) GO TO 61
   H = PCLY(2)/POLY(3)*(SQRT(Q1)/ABS(POLY(2)/POLY(3)) - 1.0)
   GC TC 63
61 H = CIR*SQRT(Q1)
   GC TC 63
62 H = - POLY(1)/POLY(2)
63 IF (ABS(F).LT.100.0*TOLR.AND.ABS(POLY(2)/POLY(1)).GT..01/TOLR)
   X GC TC 59
   IF (ABS(F).LT.1.00) GO TO 65
   IF (ABS(X).GT.0.1) GO TO 64
   H = H/ABS(H)*0.1
   GC TC 65
64 IF (H/X.GT.0.5) H = 0.5*X
   IF (H/X.LT.-1.5) H = -1.5*X
65 DELTX = F*CIR
   GC TC 68
66 IF (ITRIG.GE.15) H = 1.5*H
   DELTX = F
   ITRIG = ITRIG + 1
68 JTRIG = -1
   GC TC 102
70 IF (IREAL.EQ.0) GO TO 275

```

C --- STATEMENTS 70 THROUGH 77 INVESTIGATE Q1 LESS THE 0.

```

72 IF (ITRIG.GT.15) GO TO 110
   IF (ITRIG.GT.10.AND.ABS(P(1)).GT.1.0/TOLRQ) GO TO 110
   IF (JTRIG.GT.0.OR.ITRIG.GE.2) GO TO 74
   H1 = -PCLY(2)/POLY(3)*DIR
   IF (ITRIG.EQ.0.AND.ABS(H1).GT.1.0) H1 = H1/ABS(H1)
   H = H1
   IF (X.EQ.XI) GO TO 73
   IF (ABS(F1/X).GT.1.0) H1 = H1*ABS(X/H1)
   H = H1
   IF ((X -XI)*DIR/H.LT.0.0) H = -H
73 IF (ITRIG.LE.0) ITRIG = 1
   JTRIG = 1
   PCLY1R = 1.0
   GC TC 100
74 POLY1R = PCLY(1)/OPOLY1
   IF (JTRIG.LT.0) H1 = H
   IF (PCLY1R.LE.0.0) GO TO 75
   IF (PCLY(2)/OPOLY2.LT.0.0.AND.ITRIG.LT.2) GO TO 76

```

C --- IF POLY1R IS GREATER THAN 0.5, NO ROOT IS EXPECTED IN THIS
C --- REGION OF Q1 LESS THAN 0.

```

   IF (PCLY1R.GT.0.95) GO TO 80
   H1= H1*FACT
   GC TC 78
75 ITRIG = -1
   IF (FACT.LT.1.0) GO TO 77
   FACT = 0.5
   H = H1
   H = -H*FACT
   GC TC 90

```

```

C ---      THE SIGN OF THE SLOPE OF POLY(X) HAS CHANGED.  THE POINT OF
C --- INTEREST WAS BYPASSED.  HALF STEP BACK.

76 IF (ABS(H1).LT.TOLR) GO TO 80
77 FACT= 0.5
  H1= -H1*FACT
78 XCPLX = X
  YCPLX = SQRT(ABS(POLY(1)))
  GO TC 10C

C ---      IF STATEMENT 80 IS REACHED, POLY(X) DID NOT CROSS THE X AXIS.
C --- CONTINUE IN THE SAME DIRECTION THAT WAS STARTED EVEN THOUGH P(X)
C --- IS MOVING AWAY FROM THE X AXIS.  THE STEP SIZE IS DOUBLED UNTIL
C --- THE SECOND DERIVATIVE CHANGES SIGN.

80 IF (IREAL.EQ.0) GO TO 275
  IF (ITRIG.LT.2) H = 0.1*H
  FACT = 2.0
  ITRIG = ITRIG + 1
  H = H*2.0
90 IF (H.EQ.0.0) H = 0.1*DIR
  DELTX = H
  GO TC 102
100 DELTX = H1
102 OPCLY2 = POLY(2)
  IF (OPOLY2.EQ.0.0) OPOLY2 = 1.0E-17
104 IF (ABS(X).GT.1.0) GO TO 106
  IF (ABS(DELT X).LT.TOLRQ) GO TO 160
  GO TC 108
106 IF (ABS(DELT X/X).LT.TOLRQ) GO TO 160
108 OPCLY1 = POLY(1)
  XLAST = X
  X = X + DELTX
  IF (IMSURE.EQ.0) GO TO 30
  IF (X.LE.XHIGH.AND.X.GE.XLOW) GO TO 30
  X = (XHIGH + XLOW)/2.0
  H1 = (XHIGH - XLOW)/2.0
  IF (XHIGH.EQ.XLAST) H1 = -H1
  DELTX = H1
  GO TC 30

C ---      STATEMENT 110 IS REACHED WHEN THERE ARE NO REAL ROOTS FOUND
C --- IN THE DIRECTION STARTED.  GO BACK TO XI AND WORK THE OTHER
C --- DIRECTION IF IT HAS NOT BEEN INVESTIGATED.

110 IF (IMSURE.EQ.0) GO TO 115
  DELTX = (XHIGH - XLOW)/2.0
  IF (X.EQ.XHIGH) DELTX = -DELT X
  GO TC 102
115 IF (IDIR) 130,130,120
120 IDIR = -1.0
  X = XI
  GO TC 20

C ---      STATEMENT 130 IS REACHED IF NO REAL ROOTS WERE FOUND IN
C --- EITHER SIDE OF X = XI.  SEARCH FOR COMPLEX ROOTS.

```

```

13C IREAL = C
    X = XCPLX
    Y = YCPLX
    GC TC 28C
14C IF (ABS(F(1)).LT.0.5*TOLRQ) GO TO 142
    CPCLY1 = PCLY(1)
    XLAST = X
    IF (ITRIG.EQ.0) ITRIG = -1
142 IF (IMSURE.EQ.0) GO TO 150
    XH = X + DELTX
    IF (XH.LT.XHIGH.AND.XH.GT.XLOW) GO TO 150
    IF (LTRIG.EQ.1) GO TO 193
    IF (KTRIG.EQ.2) GO TO 1244
    IF (LTRIG.LT.0) LTRIG = LTRIG - 10
    IF (ABS(PXHIGH).LT.TOLRQ.AND.ABS(PXLOW).LT.TOLRQ) GO TO 155
    XL = 0.0
    IF (DELTX.GT.0.0) GO TO 145
143 IF (ABS(PXLOW).LT.TOLRQ) GO TO 145
    IF (X.NE.XLOW) GO TO 144
    XL = -0.5*DELTX
    IF (XHIGH.GE.X-DELTX) GO TO 144
    XL = SQRT(ABS(PXLOW))
    XL = (XHIGH - XLOW)*XL/(XL + SQRT(ABS(PXHIGH)))
144 DELTX = XL + (XLOW - X - XL)/(0.7 + 0.5*ALOG(ABS(PXLOW*2.0/TOLRQ)))
    ICHECK = -1
    GO TC 14E
145 IF (ABS(PXHIGH).LT.TOLRQ) GO TO 143
    IF (X.NE.XHIGH) GO TO 146
    XL = -0.5*DELTX
    IF (XLOW.LE.X-DELTX) GO TO 146
    XL = SQRT(ABS(PXHIGH))
    XL = (XLOW - XHIGH)*XL/(XL + SQRT(ABS(PXLOW)))
146 DELTX = XL + (XHIGH - X - XL)/(0.7 + 0.5*ALOG(ABS(PXHIGH*2.0/TOLRQ)))
    ICHECK = 1
148 IF (LTRIG.LT.0) GO TO 150
    KTRIG = C
    IXM = 0
    NC = 2
149 LTRIG = C
15C X = X + DELTX
    H1 = DELTX
    GC TC 30
152 M = M - 1
    DO 154 I=1,M
    FK = N - NCRV - I + 1
154 B(I) = B(I)*FK
    GC TC 35
155 IF (LTRIG.LT.0) GO TC 201
158 X = (XHIGH + XLOW)/2.0
    GO TC 251

```

C --- STATEMENTS 160 THROUGH 250 ARE THE RATIO OF DERIVATIVES
C --- METHOD FOR REAL ROOTS.

```

16C IF (ITRIG.GE.2) GO TO 80
    IF (ABS(F(1)).GT.TOLRQ.AND.ABS(POLY(2)/POLY(1)).LT.1.0) GO TO 165
    IF (NC.EC.NDEG.AND.LTRIG.EQ.0) GO TO 170
    IF (ABS(F(NC-1)).LT.TOLRQ) GO TO 162
    LIMIT = C
    GC TC 18C

```

```

162 LIMIT = 1
   IF (JTRIG.LT.2) GO TO 164
   IF (ABS(F(ND)).GT.TOLRQ) GO TO 180
   LIMIT = 4
   GO TO 251
164 IF (ABS(F(2)).GT.10.0*TOLR) GO TO 180
   ND = ND + 1
   GO TO 152
165 QI = (POLY(2)/POLY(3))*2 - 2.0*POLY(1)/POLY(3)
166 JTRIG = 1
   KTRIG = C
   LTRIG = C
   ND = 2
   XCPLX = X
   YCPLX = C.001
   IF (C1) 167,168,168
167 ITRIG = 1
   GO TO 80
168 IREAL = 1
   GO TO 58
170 IF (ABS(F(ND)).LE.0.5*TOLRQ) GO TO 252
   RATIO(ND) = POLY(ND+1)/POLY(ND)
   LIMIT = C
   IF (ABS(RATIO(ND)).LT.0.5/TOLR) GO TO 175
   IF (ABS(F(ND-1)).LT.TOLRQ) GO TO 176
   RATIO(ND-1) = POLY(ND)/POLY(ND-1)
   IF (ABS(RATIO(ND)/RATIO(ND-1)).GE.1.0) GO TO 180
   ND = ND + 1
   GO TO 245
175 IF (ABS(F(ND-1)).GT.0.5*TOLRQ) GO TO 180
176 DELTX = TOLR
   GO TO 140
180 IF (JTRIG.GE.2) GO TO 181
   JTRIG = 3
   CALL SCALER
   IF (NK.EQ.0) GO TO 181
   X = X/16.0C0**NK
   ITRIG = C
   JTRIG = 2
   IMSURE = 0
   GO TO 30
181 IF (ABS(F(ND-1)).LT.0.5*TOLRQ) POLY(ND-1) = 0.5*TOLRQ*CA(ND-1)
182 RATIO(ND-1) = POLY(ND)/POLY(ND-1)
185 RATIO(ND) = POLY(ND+1)/POLY(ND)
   DRATIO = RATIO(ND)/RATIO(ND-1)
   ARATIO = ABS(DRATIO)

C ---      NEAR A ROOT WITH MULTIPLICITY EQUAL TO OR GREATER THAN ND -1,
C --- THE ABSOLUTE VALUE OF ARATIO SHOULD BE LESS THAN ONE. IF ARATIO
C --- IS GREATER THAN ONE, EITHER A ROUNDOFF ERROR IS ENCOUNTERED OR WE
C --- ARE NOT RELATIVELY CLOSE TO A ROOT. IN EITHER CASE USE RATIO(ND)
C --- TO MOVE X. HOWEVER, LIMIT THE MAGNITUDE OF RATIO(ND) SO THAT THE
C --- MULTIPLICITY CHECK CAN BE MADE WITHIN COMPUTER ACCURACY.

   IF (KTRIG.EQ.1) GO TO 240
   IF (LTRIG) 200,186,190
186 IF (ARATIO.LT.1.0) GO TO 210
188 SRATIO = SQRT(ABS(RATIO(ND)*RATIO(ND-1)))
   IF (SRATIO.LT.1.0) SRATIO = 1.0

```

```

190 IF (LTRIG.EQ.1) GO TO 192
   IF (LTRIG.GT.1) GO TO 1195
   XX = X
   IXM = 1
   JTRIG = 2
   LTRIG = 1
   ODRAT = CRATIO
   DELTX = C.1/SRATIO*DIR
   DX = DELTX
   GC TC 14C
192 IF (CRATIO/ODRAT.LT.0.0) GO TO 194
   IF (ARATIO.LT.ABS(ODRAT)) GO TO 196
193 X = X - DELTX
   DELTX = -DELTX
   DX = DELTX
   GC TC 198
194 DELTX = -DELTX*ARATIO/(ARATIO + ABS(ODRAT))
   ODRAT = CRATIO
   GC TC 14C
195 IF (CRATIO/ODRAT.LT.0.0) GO TO 194
   IF (LTRIG.LT.0) GO TC 196
1195 IF (ARATIO.GT.ABS(ODRAT)) GO TO 199
196 IF (ARATIO.LE.0.45) GO TO 230
   IF (LTRIG.EQ.8.AND.IMSURE.EQ.0) GO TO 272
197 DELTX = 2.0*DELTX
   ODRAT = CRATIO
198 IF (LTRIG.GE.0) LTRIG = LTRIG + 1
   GC TC 14C
199 IF (IMSURE.EQ.0) GO TO 276
   LTRIG = -1
   IXM = 0
   GC TC 197
200 IF (LTRIG.GE.-10) GO TO 195
201 IF (LTRIG.LT.-11) GO TO 202
   LTRIG = -2
   X = XX
   DELTX = -DX
   GC TC 14C
202 IF (LTRIG.LT.-12) GO TO 204
203 IF (ND.EC.2) GO TO 158
   ND = ND - 1
   DX = (XHIGH - XLOW)/10.0
   DELTX = CX
   X = XLOW + DX
   LTRIG = -13
   GC TC 30
204 IF (ARATIO.LT.0.45) GO TO 230
205 IF (LTRIG.LT.-13) GO TO 206
   GC TC 207
206 IF (LIMIT.GT.0) GO TO 207
   IF (CRATIO/ODRAT.LT.0.0.OR.LTRIG.EQ.-15) GO TO 209
207 X = X + CX
   LTRIG = -14
   IF (X.GE.XHIGH) GO TC 203
208 ODRAT = CRATIO
   GC TC 30

```

```

205 DELTX = DELTX*DRATIO/(ODRAT - DRATIO)
   LTRIG = -15
   ODRAT = CRATIO
   GO TC 150
210 IF (JTRIG.EQ.4) GO TO 212
   IF (ARATIO - 0.45) 230,230,215
212 JTRIG = 2
   ND = ND - 1
   GO TC 30
215 IF (LIMIT.EQ.0.OR.JTRIG.EQ.3) GO TO 220
   GO TC 180
218 JTRIG = 4
220 IF (ND.EQ.NDEG) GO TO 225
   ND = ND + 1
   GO TC 152
225 LTRIG = 0
   DELTX = -1.0/RATIO(ND)
   IF (JTRIG.EQ.4) DELTX = -1.0/RATIO(ND-1)
   JTRIG = 2
   GO TC 140
226 IF (LIMIT.EQ.1) GO TO 248
   IF (LTRIG.LT.0) GO TO 229
228 ND = ND - 1
   GO TC 30
229 DELTX = CX
   GO TC 207

```

C --- RATIO(ND-1) IN STATEMENT 230 IS THE LAST DERIVATIVE RATIO
C --- THAT SHOULD APPROACH INFINITY AS X APPROACHES A ROOT VALUE.
C --- THE PROXIMITY OF X TO THE ROOT IS 1.0/RATIO(ND-1).

C --- THE MULTIPLICITY OF ROOTS AND THE ESTIMATE OF THE NEXT ROOTS
C --- ARE MADE WHEN THE POLYNOMIAL DERIVATIVE RATIO LIES BETWEEN 0.5TOLR
C --- AND 5.0TCLR. THIS MEANS THAT ROOTS CLOSER THAN TOLR TOGETHER
C --- CANNOT BE RESOLVED FROM ONE ANOTHER. THE ACTUAL X VALUE OF THE
C --- RCCT, HOWEVER, IS IMPROVED SOMEWHAT IN STATEMENT 250.

```

230 IF (LTRIG.GT.0) LTRIG = 0
   IF (ND.EQ.2) JTRIG = 2
   IF (JTRIG.EQ.3) GO TO 218
   IF (ABS(RATIO(ND-1)).LT.5.0/TOLR) GO TO 235
232 IF (ABS(RATIO(ND-1)).LT.1.0/TOLRQ) GO TO 233
   DELTX = 1.0/(TOLR*RATIO(ND-1))
   GO TC 150
233 DELTX = -0.2*TOLR
   GO TC 150
235 IF (ABS(RATIO(ND-1)).GT.1.0/TOLR) GO TO 242
   IF (LIMIT.GT.0) GO TO 238
   DELTX = -1.0/RATIO(ND-1)
   GO TC 140
238 IF (ABS(RATIO(ND-1)).LT.1000.0) GO TO 251
   XCLC = X
   ORATIC = RATIO(ND-1)
   KTRIG = 1
   DELTX = 0.1/ORATIO
   CRATIC = SQRT(ABS(ORATIO))
   GO TC 150

```

```

240 IF (ARATIO.GE.0.45) GO TO 241
    IF (ABS(RATIO(ND-1)).LT.0RATIO) GO TO 241
    IF (LIMIT.EQ.0) GO TO 248
    DELTX = 2.0*DELT
    GC TC 140
241 X = XOLD
    GC TC 251
242 IF (ND.LE.2) GO TO 245
    IF (LIMIT.GT.0) GO TO 1245
    IF (ABS(P(ND-2)).GE.0.5*TOLRQ) GO TO 243
    LIMIT = 1
    PCLY(ND-2) = 0.5*TOLRQ*CA(ND-2)
243 RATIO(ND-2) = POLY(ND-1)/POLY(ND-2)
    ARATIO = ABS(RATIO(ND-1)/RATIO(ND-2))
    IF (KTRIG.EQ.2) GO TO 244
    IF (ARATIO.LT.0.53) GO TO 248
    IF (IXM.EQ.0) GO TO 226
    CARAT = ARATIO
    XPN = X
    KTRIG = 2
    GC TC 247
244 IF (ARATIO.LT.0.53) GO TO 248
    IF (ARATIO.GT.ABS(ODRAT)) GO TO 1244
    IF (LIMIT.EQ.0) GO TO 228
    GC TC 248
1244 X = XPN
    KTRIG = 0
    IXM = 0
    DELTX = 0.0
    GC TC 145

C ---      THE MULTIPLICITY OF THE ROOT AND ITS VALUE WITHIN TOLR ARE
C --- NOW AT HAND.  ESTIMATE THE X LOCATION OF THE NEXT ROOT.

245 IF (ABS(POLY(3)).GT.ABS(POLY(1))) GO TO 248
1245 IF (IXM.EQ.0) GO TO 248
246 IXM = 0
247 DELTX = 2.0*(XX - X)
    LTRIG = 0
    GC TC 140
248 MULT = NC - 1
    XMULT = MULT
    IF (NDEG - MULT.LT.3) GO TO 250
    RA = 1.0/(XMULT + 1.0)*RATIO(ND)
    XI = X - 1.0/RA
    IF (ABS(RA).LT.0.1) XI = X
250 X = X - 1.0/RATIO(MULT)
    GC TC 255
251 NC = 1
252 MULT = NC
    XI = X

C ---      NUMER THE ROOTS AND DIVIDE THEM OUT OF THE POLYNOMIAL.

255 DC 270 K=1,MULT
    J = J + 1
    XRCCT(J) = X*SCALE
    YRCCT(J) = 0.0

```

```

      N = N-1
      IF (N.EQ.1) GO TO 265
      DC 260 I=2,N
260  A(I) = A(I) + X*A(I-1)
265  XR(J) = 1.0 + X*A(N)/A(N+1)
270  YR(J) = C.0
      GC TC 19
272  X = XX
      GC TC 278
275  IREAL = -1
276  X = X - DELTX
278  Y = SQRT(ABS(POLY(1)))
280  CALL COMPLX
      IF (IREAL.LT.2) GO TO 19
      IREAL = C
      GC TC 252
570  CALL QED
1000 FORMAT (I3/(3D24.16))
1010 FORMAT (//25X,78HERROR ---- THE DEGREE OF THE POLYNOMIAL IS LESS T
      XHAN CNE. THERE ARE NO ROOTS. )
1020 FORMAT (//37X,55HERROR --- THE PROGRAM IS ONLY DIMENSIONED FOR N =
      X 100. )
1030 FORMAT (1H1///31X,70HTHE INPUT POLYNOMIAL COEFFICIENTS. THEY ARE
      X IN ORDER IF READ IN ROWS. // (2X,5D24.16))
      GC TC 5
      ENC

```

SUBROUTINE RPOLY

```

C ---      THIS IS THE BASIC ROUTINE FOR FINDING THE VALUE OF A REAL
C ---      NUMBER PCLYNOMIAL AND ITS DERIVATIVES. P( ) ARE THE POLYNOMIAL
C ---      AND ITS DERIVATIVE VALUES NORMALIZED BY B(N).

      DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
      X RA, RATIO, RRATIO, SCALE, X, XLAST, XP, XPN, XPOLY, XR,
      X XRCCT, Y, YP, YPOLY, YR, YROOT
      COMMON /POLYN/ ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
      1 ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
      2 NDEG, NCRV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,
      3 TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
      4 A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
      5 RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
      6,YP(100), YPOLY(100), YR(100), YROOT(100)
35  P(NCRV+1) = 1.0
      L = N
      IF (P.EQ.0) GO TO 45
      NC = NEXPS - NEXP
      IF (L.EQ.N-1) GO TO 37
      NM = NC
36  IF (ABS(E(L+1)/CA(NDRV)).GT.0.001*8.0**NM*TOLRQ) GO TO 37
      L = L - 1
      IF (L.EQ.0) GO TO 37
      NM = NM + NC
      GC TC 36

```



```

37 I = 1
38 IF (I.EQ.L+1) GO TO 42
   P(NCRV+1) = P(NDRV+1)*B(I)
   II = I
   NM = NC
39 IF (ABS(E(I+1)/B(II)).GT.0.001*8.0**NM*TOLRQ) GO TO 40
   P(NCRV+1) = P(NDRV+1)*X
   IF (I.EQ.L) GO TO 41
   I = I + 1
   NM = NM + NC
   GC TC 39
40 P(NCRV+1) = P(NDRV+1)*X/B(I+1) + 1.0
   I = I + 1
   GC TC 38
41 P(NCRV+1) = P(NDRV+1)/B(I+1) + 1.0
42 IF (L.EQ.M) GO TO 45
   P(NCRV+1) = P(NDRV+1)*X**(M-L)
45 POLY(NDRV+1) = P(NDRV+1)*B(L+1)
   CA(NCRV+1) = B(L+1)
   NDRV = NCRV + 1
   IF (NCRV.GT.ND) RETURN
48 M = M-1
   DC 49 I=1,M
   FK = N - NDRV - I + 1
49 B(I) = B(I)*FK
   GC TC 35
   END

```

SUBROUTINE COMPLX

C --- THIS SUBROUTINE CONTAINS THE ANALYSIS PROCEDURES FOR COMPLEX
C --- ROOTS.

```

DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
X RA, RATIO, RRATIO, SCALE, X, XLAST, XP, XPN, XPOLY, XR,
X XRCCT, Y, YP, YPOLY, YR, YRCOT
COMMON /POLYN/ ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
1 ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
2 NDEG, NCRV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,
3 TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
4 A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
5 RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
6, YP(100), YPOLY(100), YR(100), YROOT(100)

```

C --- THE FIRST X ESTIMATE FOR THE FIRST COMPLEX ROOT IS SOME LOCAL
C --- MINIMUM OF THE ABSOLUTE VALUE OF POLY(X). LET THE 1ST Y ESTIMATE
C --- EQUAL THE SQUARE ROOT OF POLY(X).

```

DTHETA = 3.1415927/6.0
280 IF (IREAL.EQ.-1) GO TO 528
   IF (Y.LT.0.001) Y = 0.001
   IF (Y.GT.10.0) Y = 10.0
   IF (ABS(X).LT.0.01) X = 0.01
   NDEG = N - 1

```

```

29C JTRIG = 1
    ITRIG = C
    IXM = 0
    ND = 2
295 LTRIG = C
30C DC 310 I=1,N
31C B(I)= A(I)
    M= NDEG
    NDRV= 0
32C CALL CPOLY
351 IF (JTRIG.GE.2) GO TO 360
    IF (ABS(XP(1)).LT.TOLI.AND.ABS(YP(1)).LT.TOLI) GO TO 360

C ---      NEW X AND Y VALUES ARE DETERMINED WITH A 3 TERM COMPLEX
C ---      TAYLORS SERIES.

    XQ = (XPCLY(2)/B(N))**2 - (YPOLY(2)/B(N))**2 + 2.0*(YP(1)*
X YPCLY(3) - XP(1)*XPOLY(3))/B(N)
    YQ = 2.0*(XPOLY(2)*YPOLY(2)/B(N) - (XP(1)*YPOLY(3) + YP(1)*
X XPCLY(3)))/B(N)
    PHI = ATAN2(YQ,XQ)
    IF (PHI.LT.0.0) PHI = 6.2831853 + PHI
    PHI = PHI/2.0
    HYP = (XQ**2 + YQ**2)**0.25
    XQ = HYP*CCS(PHI)
    YQ = HYP*SIN(PHI)
    DXT = XPCLY(2)/B(N)
    DYT = YPCLY(2)/B(N)
    DZT = (XQ - DXT)**2 + (YQ - DYT)**2
    DZT2 = (XQ + DXT)**2 + (YQ + DYT)**2
    IF (DZT.LE.DZT2) GO TO 352
    XQ = -XQ
    YQ = -YQ
    DZT = DZT2
352 XNT = XQ - DXT
    YNT = YQ - DYT
    XDT = XPCLY(3)/B(N)
    YDT = YPCLY(3)/B(N)
    D = XDT**2 + YDT**2
    HX = (XNT*XDT + YNT*YDT)/D
    HY = (YNT*XDT - XNT*YDT)/D
    IF (ABS(HX/X).LT.0.1*TOLI.AND.ABS(HY/Y).LT.0.1*TOLI) GO TO 360
    IF (ABS(HX).LT.1.0) GO TO 357
    IF (ABS(X).GT.0.1) GO TO 356
    HX = HX/ABS(HX)*0.1
    GC TC 357
356 IF (ABS(HX/X).GT.0.5) HX = 0.5*HX*ABS(X/HX)
357 IF (ABS(HY).LT.1.0) GO TO 359
    IF (ABS(Y).GT.0.1) GO TO 358
    HY = HY/ABS(HY)*0.1
    GC TC 359
358 IF (ABS(HY/Y).GT.0.5) HY = 0.5*HY*ABS(Y/HY)
359 X = X + HX
    Y = Y + HY
    GC TC 30C

C ---      STATEMENTS 360 THROUGH 535 COVER THE RATIO OF DERIVATIVES
C ---      METHOD FOR COMPLEX ROOTS.

```

```

36C IF (ABS(XP(ND-1)).LT.0.5*TOLIQ.AND.ABS(YP(ND-1)).LT.0.5*TOLIQ)
X GC TC 370
LIMIT = C
GC TC 38C

```

C --- WHEN LIMIT = 1, POLY(ND-1) HAS ROUNDOFF LIMITATIONS.

```

37C LIMIT = 1
IF (JTRIG.EQ.1) GO TO 375

```

C --- WHEN LIMIT = 4, POLY(ND) HAS GREATER ROUNDOFF LIMITATIONS.

```

IF (ABS(XP(ND)).GT.0.5*TOLIQ.OR.ABS(YP(ND)).GT.0.5*TOLIQ) GO TO 380
LIMIT = 4
GC TC 512
375 IF (ND.GT.NDEG/2) GO TO 380
ND = ND + 1
GC TC 482
38C IF (JTRIG.EQ.2) GO TO 382
CALL SCALER
IF (NK.EQ.0) GO TO 382
X = X/16.0C0**NK
Y = Y/16.0C0**NK
JTRIG = 2
GC TC 30C
382 IF (ABS(XP(ND-1)).LT.0.5*TOLIQ) XP(ND-1) = 0.5*TOLIQ
IF (ABS(YP(ND-1)).LT.0.5*TOLIQ) YP(ND-1) = 0.5*TOLIQ
D = (XP(ND-1))**2 + (YP(ND-1))**2
PCLY(ND-1) = DSQRT(D)*DABS(CA(ND-1))
IF (ABS(XP(ND)).LT.0.5*TOLIQ) XP(ND) = 0.5*TOLIQ
IF (ABS(YP(ND)).LT.0.5*TOLIQ) YP(ND) = 0.5*TOLIQ
D1 = (XP(ND))**2 + (YP(ND))**2
PCLY(ND) = DSQRT(D1)*DABS(CA(ND))
RATIC(ND-1) = POLY(ND)/POLY(ND-1)
IF (ND.LE.2.OR.LIMIT.GT.0) GO TO 385
IF (ABS(XP(ND-2)).LT.0.5*TOLIQ) XP(ND-2) = 0.5*TOLIQ
IF (ABS(YP(ND-2)).LT.0.5*TOLIQ) YP(ND-2) = 0.5*TOLIQ
PCLY(ND-2) = DSQRT((XP(ND-2))**2 + (YP(ND-2))**2)*DABS(CA(ND-2))
RATIC(ND-2) = POLY(ND-1)/POLY(ND-2)
IF (LTRIG.NE.0) GO TO 385
ARATIC = RATIO(ND-1)/RATIO(ND-2)
IF (ARATIO.LT.0.45) GO TO 440
IF (ABS(XP(ND-2)).LT.0.6*TOLIQ.AND.ABS(YP(ND-2)).LT.0.6*TOLIQ)
X GC TC 385
IF (ARATIO.GT.1.0) GO TO 455
385 PCLY(ND+1) = DSQRT((XPOLY(ND+1))**2 + (YPOLY(ND+1))**2)
RATIC(ND) = POLY(ND+1)/POLY(ND)
ARATIC = RATIO(ND)/RATIO(ND-1)
RRATIC(ND-1) = (XP(ND)*XP(ND-1)+YP(ND)*YP(ND-1))/D*CA(ND)/CA(ND-1)
CRATIO(ND-1) = (YP(ND)*XP(ND-1)-XP(ND)*YP(ND-1))/D*CA(ND)/CA(ND-1)
D3 = (RRATIC(ND-1))**2 + (CRATIO(ND-1))**2
RRATIC(ND) = (XP(ND+1)*XP(ND)+YP(ND+1)*YP(ND))/D1*CA(ND+1)/CA(ND)
CRATIO(ND) = (YP(ND+1)*XP(ND)-XP(ND+1)*YP(ND))/D1*CA(ND+1)/CA(ND)
D2 = (RRATIC(ND))**2 + (CRATIO(ND))**2
JTRIG = 2
IF (LTRIG) 389,387,395
387 IF (LTRIG.EQ.1) GO TO 505
IF (ARATIO.LT.1.0) GC TC 460

```

C --- IF ARATIO IS GREATER THAN ONE WHERE LIMIT = 1, THE PRESENT
 C --- POINT VERY LIKELY IS BETWEEN TWO CLOSE ROOTS. THE FOLLOWING
 C --- STATEMENTS STEP IN A SPIRAL TO LOWER ARATIO. IF ARATIO IS
 C --- GREATER THAN ONE WHERE LIMIT = 1, POSITIVE ANALYSIS IS NOT
 C --- POSSIBLE. IN AN ATTEMPT TO ANALYZE WITHOUT THE LIMIT SOME STEPS
 C --- ARE TAKEN IN THIS CASE TOO. HOWEVER, IT IS USUALLY NECESSARY TO
 C --- DIVIDE OUT ONE ROOT AT THE BEST COORDINATES WHEN LIMIT = 1.

```

389 IF (ARATIO.LT.0.45) GO TO 490
390 SRATIO = SQRT(RATIO(ND)*RATIO(ND-1))
    IF (SRATIO.LT.1.0) SRATIO = 1.0
395 IF (LTRIG.EQ.1) GO TO 405
    IF (LTRIG.EQ.2) GO TO 410
    IF (LTRIG.GT.2) GO TO 415
    XX = X
    YY = Y
400 LTRIG = 1
    ISIGN = C
    OAX = ARATIO
    X = X + C.1/SRATIO
    GO TO 300
405 LTRIG = 2
    OAY = ARATIO
    X = X - C.1/SRATIO
    Y = Y + C.1/SRATIO
    GO TO 300
410 Y = Y - C.1/SRATIO
    XCCN = (CAX - OAY)/OAX
    YCCN = (CAX - ARATIO)/OAX
    ZCCN = SQRT(XCON**2 + YCON**2)
    THETA0 = ATAN2(YCON,XCON) - CTHETA
    XPCLE = X
    YPCLE = Y
411 THETA = CTHETA
    IF (ZCON.GT.1.414) SRATIO = 10.0*SRATIO
    RAD = 0.6*ABS(THETA)/SRATIO
    DELTX = RAD*COS(THETA + THETA0)
    DELTY = RAD*SIN(THETA + THETA0)
412 CARAT = CAX
413 XLAST = X
    YLAST = Y
    X = XPCLE + DELTX
    Y = YPCLE + DELTY
    LTRIG = LTRIG + 1
    GO TO 300
415 IF (ARATIO.LE.0.45) GO TO 490
    IF (ARATIO.GT.OARAT) GO TO 420
    IF (LTRIG.NE.3.OR.ARATIO.GE.1.0) GO TO 418
    IF (SRATIO.LT.1000.0.OR.ARATIO.LT.OARAT-0.01) GO TO 418
    LTRIG = C
    GO TO 480
418 IF (LTRIG.GT.10) GO TO 425
    THETA = THETA + CTHETA
    RAD = 0.6*ABS(THETA)/SRATIO
    DELTX = RAD*COS(THETA + THETA0)
    DELTY = RAD*SIN(THETA + THETA0)
    CARAT = ARATIO
    GO TO 413
  
```

```

42C IF (LIMIT.GT.0.OR.LTRIG.EQ.3) GO TO 425
    X = XLAST
    Y = YLAST
    DTHETA = -DTHETA
    LTRIG = -1
    GC TC 30C
425 X = XX
    Y = YY
    IF (LIMIT.GT.0) GO TO 512
    LTRIG = 2
    ISIGN = ISIGN + 1
    IF (ISIGN - 2) 430,432,434
43C THETA0 = THETA0 - 3.0*DTHETA
    GC TC 411
432 THETA0 = THETA0 + 6.0*DTHETA
    GC TC 411
434 IF (ISIGN.LE.3) GO TC 438
    THETA0 = THETA0 - 6.0*DTHETA
    SRATIO = 10.0*SRATIO
    GC TC 411
438 THETA0 = THETA0 + 3.0*DTHETA
    GC TC 411
44C ND = ND - 1
    LIMIT = 1
    IF (JTRIG.EQ.1) GO TC 450
    IF (ABS(XP(ND-1)).LT.0.6*TOLIQ.AND.ABS(YP(ND-1)).LT.0.6*TOLIQ)
X GC TC 450
    LIMIT = C
    D = (XP(ND-1))**2 + (YP(ND-1))**2
    RRATIO(ND-1) = (XP(ND)*XP(ND-1)+YP(ND)*YP(ND-1))/D*CA(ND)/CA(ND-1)
    CRATIO(ND-1) = (YP(ND)*XP(ND-1)-XP(ND)*YP(ND-1))/D*CA(ND)/CA(ND-1)
    D3 = (RRATIO(ND-1))**2 + (CRATIO(ND-1))**2
    X = X - RRATIO(ND-1)/D3
    Y = Y + CRATIO(ND-1)/D3
45C JTRIG = 2
    GC TC 30C
455 ND = ND - 1
    JTRIG = 2
    GC TC 39C
46C IF (ARATIO - 0.45) 490,490,470
47C IF (LIMIT.EQ.0) GO TO 480
    GC TC 39C
48C IF (ND.GT.NDEG/2) GO TO 498
    ND = ND + 1
482 N = N - 1
    DC 484 I=1,N
    FK = N - NCRV - I + 1
484 B(I) = B(I)*FK
    GC TC 32C
49C IF (RATIO(ND-1).LT.5.0/TOLI) GO TO 495
    X = X + C.2*TOLI
    Y = Y + C.2*TOLI
    GC TC 30C
495 LTRIG = C
    IF (RATIO(ND-1).GT.1.0/TOLI) GO TO 520
    IF (LIMIT.GT.0) GO TC 500
498 X = X -RRATIO(ND-1)/D3
    Y = Y + CRATIO(ND-1)/D3
    GC TC 30C

```

```

50C IF (ABS(RATIO(ND-1)).LT.1000.0) GO TO 512
XCLC = X
YCLC = Y
ORATIO = RATIO(ND-1)
ITRIG = 1
DELT = 0.1/ORATIO
IF (Y.LT.0.0) DELT = -DELT
Y = Y + DELT
GO TO 30C
505 IF (ARATIO.GE.0.45) GO TO 510
IF (LIMIT.EQ.0) GO TO 520
IF (ARATIO.GT.SQRT(ORATIO)) GO TO 510
DELT = 2.0*DELT
Y = Y + DELT
GO TO 30C
51C X = XCLC
Y = YCLC

```

C --- CHECK BACK TO SEE IF ANOTHER REAL ROOT HAS BEEN FOUND.

```

512 IF (ABS(Y).GT.10.0*TOLI) GO TO 515
NC = 1
IREAL = 2
RETURN

```

C --- DIVIDE OUT ONLY ONE ROOT SINCE THE ROOT POINT COULD NOT BE
C --- LOCATED AS WELL AS DESIRED.

```

515 MULT = 1
XI = X
YI = Y
GO TO 54C
52C IF (POLY(ND-1).LT.POLY(ND+1)) GO TO 525
IF (ND.GT.2) GO TO 523
NC = ND + 1
LTRIG = -1
GO TO 482
523 ND = ND - 1
GO TO 30C
525 IF (ABS(Y).GT.10.0*TOLI) GO TO 530
IREAL = C
XX = X
NND = ND
RETURN
528 X = XX
NC = NND
IREAL = C

```

C --- THE MULTIPLICITY OF THE ROOT AND ITS VALUE WITHIN TOLI ARE
C --- NOW AT HAND. ESTIMATE THE Z LOCATION OF THE NEXT CLOSEST ROOT
C --- THAT IS NOT THE COMPLIMENT OF THE PRESENT ROOT.

```

53C MULT = ND-1
XMULT = MULT
IF (NDEG - 2*MULT.LT.3) GO TO 535
RATIOR = FRATIO(ND)/(XMULT + 1.0)
RATIOI = CRATIO(ND)/(XMULT + 1.0) + XMULT/(2.0*Y)
OX = RATIOR*Y/(Y*(RATIOR**2 + RATIOI**2) + RATIOI)

```

```

      XI = X - DX
      YI = SQRT(ABS((RATIOI*Y + RATIOI*DX)*(DX*DX+ Y*Y)/(RATIOI*Y -
X RATIOI*DX)))
535 D2= (RRATIO(MULT))**2 + (CRATIO(MULT))**2
      X= X - RRATIO(MULT)/D2
      Y= Y + CRATIO(MULT)/D2

C ---      NUMER THE ROOTS AND DIVIDE THEM OUT OF THE POLYNOMIAL FIRST
C --- AS COMPLEX NUMBERS TO FIND THE COMPLEX RESIDUAL. THEN GO BACK AND
C --- DIVIDE OUT THE ROOT AND ITS COMPLEMENT TOGETHER SO THAT ONLY REAL
C --- NUMBER ALGEBRA IS USED.
540 DC 565 K=1,MULT
      DC 545 I=1,N
      RA(I) = A(I)
545 CA(I) = C.0
      J= J+1
      XRCCT(J) = X*SCALE
      YRCCT(J) = Y*SCALE
      N= N-1
      DC 550 I=2,N
      XPN= RA(I) + X*RA(I-1) - Y*CA(I-1)
      CA(I)= CA(I) + X*CA(I-1) + Y*RA(I-1)
550 RA(I)= XPN
      XR(J) = (RA(N+1) + X*RA(N) - Y*CA(N))/A(N+1)
555 YR(J) = (CA(N+1) + X*CA(N) + Y*RA(N))/A(N+1)
      J= J+1
      XRCCT(J) = X*SCALE
      YRCCT(J) =-Y*SCALE
      N = N - 1
      A(2)= A(2) + 2.0*A(1)*X
      IF (N.EQ.1) GO TO 562
      K= N+1
      DC 560 I=3,K
560 A(I)= A(I) + 2.0*A(I-1)*X - A(I-2)*(X**2 + Y**2)
562 XR(J) = 1.0 + (A(N+1)*X - A(N)*(X**2 + Y**2))/A(N+2)
565 YR(J)= 0.0
      X= XI
      Y= YI
      IF (X.GT.10.0) X = 10.0
      IF ((N+1)/2.EQ.N/2) IREAL = 1
      RETURN
      END

```

SUBROUTINE CPOLY

```

C ---      THIS IS THE BASIC ROUTINE FOR FINDING THE VALUE OF THE
C --- POLYNOMIAL AND ITS DERIVATIVES IN COMPLEX NUMBERS.

```

```

      DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
X RA, RATIO, RRATIO, SCALE, X, XLAST, XP, XPN, XPOLY, XR,
X XRCCT, Y, YP, YPOLY, YR, YRCOT
      COMMON /POLYN/ ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
1 ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
2 NCEG, NCRV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,

```

```

3 TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
4 A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
5 RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
6, YP(100), YPOLY(100), YR(100), YROOT(100)
32C XP(NDRV+1)= 1.0
    YP(NDRV+1)= 0.0
    L= M
    IF (M.EQ.0) GO TO 345
    NC = NEXFS - NEXP
    IF (L.EQ.N-1) GO TO 328
    NM = NC
325 IF (ABS(E(L+1)/CA(NDRV)).GT.0.00001*8.0**NM*TOLIQ) GO TO 328
    L= L-1
    IF (L.EQ.0) GO TO 340
    NM = NM + NC
    GO TC 325
328 I = 1
33C IF (I.EQ.L+1) GO TO 340
    XP(NDRV+1) = XP(NDRV+1)*B(I)
    II = I
    NM = NC
332 IF (ABS(E(I+1)/B(II)).GT.0.00001*8.0**NM*TOLIQ) GO TO 335
    XPN= XP(NDRV+1)*X - YP(NDRV+1)*Y
    YP(NDRV+1)= XP(NDRV+1)*Y + YP(NDRV+1)*X
    XP(NDRV+1)= XPN
    IF (I.EQ.L) GO TO 338
    I = I + 1
    NM = NM + NC
    GO TC 332
335 XPN = XP(NDRV+1)*X - YP(NDRV+1)*Y
    YP(NDRV+1) = XP(NDRV+1)*Y + YP(NDRV+1)*X
    XP(NDRV+1) = XPN/B(I+1) + 1.0
    I = I + 1
    GO TC 33C
338 XP(NDRV+1) = XP(NDRV+1)/B(I+1) + 1.0
34C IF (L.EQ.M) GO TC 345
    YP(NDRV+1) = YP(NDRV+1)/B(L+1)
    ML = M - L
    DC 342 I=1,ML
    XPN = XP(NDRV+1)*X - YP(NDRV+1)*Y
    YP(NDRV+1) = XP(NDRV+1)*Y + YP(NDRV+1)*X
342 XP(NDRV+1) = XPN
    YPCLY(NDRV+1) = YP(NDRV+1)*B(L+1)
    GO TC 346
345 YPCLY(NDRV+1) = YP(NDRV+1)
    YP(NDRV+1) = YP(NDRV+1)/B(L+1)
346 XPCLY(NDRV+1) = XP(NDRV+1)*B(L+1)
    CA(NDRV+1) = B(L+1)
    NDRV= NDRV+1
    IF (NDRV.GT.ND) RETURN
348 M= M-1
    DC 350 I=1,M
    FK= M-NDRV-I + 1
35C B(I)= B(I)*FK
    GO TC 32C
END

```


SUBRCUTINE SCALER

C --- THIS SUBROUTINE SCALES THE POLYNOMIAL SO THE LOCAL ROOT IS
C --- OF ORDER ONE.

```

DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
X RA, RATIO, RRATIO, SCALE, X, XLAST, XP, XPN, XPOLY, XR, XROOT, Y,
X YP, YPCLY, YR, YROOT
CCMPCN /POLYN/      ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
1 ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
2 NDEG, NCRV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,
3 TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
4 A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
5 RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
6, YP(100), YPOLY(100), YR(100), YROOT(100)
NK = 0
INK = 1
IF (IREAL.LE.0) GO TO 10
XPN = 0.36067376*ALOG(ABS(X))
GO TO 20
10 XPN = 0.18033688*ALOG(X**2 + Y**2)
20 IF (XPN.GT.0.0) GO TO 30
XPN = -XPN
INK = -1
30 DO 40 I=1,6
IF (XPN.LT.0.5.OR.IABS(NC + (NDEG*(NK + INK))/2).GT.20) GO TO 50
NK = NK + INK
40 XPN = XPN - 1.0
50 IF (NK.EQ.0) RETURN
NC = (NDEG*NK)/2
NEXP = NEXP + NK
SCALE = 16.0D0**NEXP
DO 60 I=1,N
NM = NK*(I-1) - NC
60 A(I) = A(I)/16.0D0**NM
RETURN
END

```

SUBRCUTINE QED

C --- IN THIS SUBROUTINE THE LAST TWO ROOTS ARE CALCULATED DIRECTLY
C --- AND ALL THE ROOTS ARE PRINTED.

```

DOUBLE PRECISION A, ARATIO, B, CA, CRATIO, D, D2, FK, P, POLY, Q,
X RA, RATIO, RRATIO, SCALE, X, XLAST, XP, XPN, XPOLY, XR,
X XRCCT, Y, YP, YPOLY, YR, YRCOT
CCMPCN /POLYN/      ARATIO, D, DELTX, DX, D2, FK, I, INK, IREAL,
1 ITRIG, IW, J, JTRIG, L, LIMIT, LTRIG, M, ML, MULT, N, NC, ND,
2 NDEG, NCRV, NEXP, NEXPS, NK, NM, NUM, OARAT, Q, SCALE, SRATIO,
3 TCLI, TCLIQ, TOLR, TOLRQ, X, XI, XLAST, XMULT, XOLD, XPN, Y,
4 A(100), B(100), CA(100), CRATIO(100), P(100), POLY(100), RA(100),
5 RATIO(100), RRATIO(100), XP(100), XPOLY(100), XR(100), XROOT(100)
6, YP(100), YPOLY(100), YR(100), YROOT(100)
57C IF (N-2) 600,575,580

```

```

575 X = -A(2)/A(1)
    J = J + 1
    XRCCT(J) = X*SCALE
    YRCCT(J) = 0.0
    XR(J) = C.0
    YR(J) = C.0
    GC TC 60C

C ---      WHEN THERE ARE ONLY TWO ROOTS LEFT, IT IS EASIER TO SOLVE FOR
C ---      THEM DIRECTLY.

58C Q = A(2)**2 - 4.0*A(1)*A(3)
    IF (C.LT.0.0) GO TO 590
    Q = DSQRT(Q)
    J = J + 1
    XRCCT(J) = (Q - A(2))/(2.0*A(1))*SCALE
    YRCCT(J) = 0.0
    XR(J) = C.0
    YR(J) = C.0
    J = J + 1
    XRCCT(J) = -(Q + A(2))/(2.0*A(1))*SCALE
    YRCCT(J) = 0.0
    XR(J) = C.0
    YR(J) = C.0
    GC TC 60C

59C J = J + 1
    XRCCT(J) = -A(2)/(2.0*A(1))*SCALE
    YRCCT(J) = DSQRT(-Q)/(2.0*A(1))*SCALE
    XR(J) = C.0
    YR(J) = C.0
    J = J + 1
    XRCCT(J) = XROOT(J-1)
    YRCCT(J) = -YROOT(J-1)
    XR(J) = C.0
    YR(J) = C.0

60C WRITE (IW,1040)
    DC 61C J=1,NUM

61C WRITE (IW,1050) J, XROOT(J), YROOT(J), XR(J), YR(J)
    WRITE (IW,1060)

104C FORMAT (//////// 39X, 54(1H*)/39X,1H*, 52X,1H*/39X,1H*,4X,48H*** THE
1  CCOMPUTED ROOTS OF THE POLYNOMIAL ***      ,1H*/39X,1H*,52X,1H*/13X,
2  106(1H*)/13X,1H*,8X,1H*,47X,1H*,47X,1H*/13X,1H*,8X,1H*,47X,1H*,
3  11X,25HPOLYNOMIAL REMAINDER WHEN,11X,1H*/13X,10H*  ROOT  *,8X,
4  9HREAL PART,13X,9HIMAGINARY,8X,1H*,13X,21HROOT WAS DIVIDED OUT ,
5  13X,1H*/13X,1H*,1X,7HNUMBER ,1H*,9X,7HOF ROOT,13X,12HPART OF ROOT
6  ,6X,1H*,47X,1H*/13X,1H*,8X,1H*,47X,1H*,8X,9HREAL PART,11X,14HIMAG
7  INARY PART,5X,1H*/13X,1H*,8X,1H*,47X,1H*,47X,1H*/13X,106(1H*)/13X,
8  1H*,8X,1H*,47X,1H*,47X,1H*)

105C FORMAT (13X,1H*,2X,I3,3X,1H*,2D22.12,3X,1H*,2D22.12,3X,1H*)
106C FORMAT (13X,1H*,8X,1H*,47X,1H*,47X,1H*/13X,106(1H*))
    RETURN
    ENC

```

Examples

THE INPUT POLYNOMIAL COEFFICIENTS. THEY ARE IN ORDER IF READ IN ROWS.

```
0.100000000000000000 01 -0.499986999999999999 02 0.999960000000000000 03 -0.999939999999999999 04 0.499960000000000000 05
-C.999900000000000000 05
```

```

*****
*                                     *
*      *** THE COMPUTED ROOTS OF THE POLYNOMIAL ***      *
*                                     *
*****

*          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *
* RCOT     REAL PART    IMAGINARY    POLYNOMIAL REMAINDER WHEN
* NUMBER   OF ROOT     PART OF ROOT  ROOT WAS DIVIDED OUT
*          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *
*****

*          *          *          *          *          *          *          *
* 1         0.100000C00005D 02      0.        0.888178419700D-15      0.
* 2         0.100000C00005D 02      0.       -0.122124532709D-14      0.
* 3         0.100000C00005D 02      0.       -0.999200722163D-15      0.
* 4         0.100000C00005D 02      0.        0.205391259556D-13      0.
* 5         0.999899999812D 01      0.         0.         0.
*          *          *          *          *          *          *          *
*****

```

Exact polynomial:

$$P(x) = (x - 10)^4(x - 9.999)$$

Comment: This problem illustrates the point that good resolution capability is maintained near multiple roots when the multiple root in a group is found first.

```
0.100000000000000000D 01 -0.499990000000000000D 01 0.899996000000000000D 01 -0.699995000000000000D 01 0.199998000000000000D 01
```

```
*****  
*                                     *  
*   *** THE COMPUTED ROOTS OF THE POLYNOMIAL ***   *  
*                                     *  
*****
```

* RCOT *		* REAL PART		* IMAGINARY		* POLYNOMIAL REMAINDER WHEN		*
* NUMBER *		* OF ROOT		* PART OF ROOT		* ROOT WAS DIVIDED OUT		*
*		*		*		* REAL PART		* IMAGINARY PART
*		*		*		*		*
* 1 *	@	0.99999999980D 00		0.		-0.		0.
* 2 *	*	0.99999999980D 00		0.		-0.666133814775D-15		0.
* 3 *	*	0.200000000000D 01		0.		0.		0.
* 4 *	*	0.999990000039D 00		0.		0.		0.
*		*		*		*		*

Comment: This problem again shows good resolution near a multiple root. However, in this case the decision between $x = 1.000000$ and $x = 0.999993$ for the location of the multiple root was barely possible or maybe even lucky. When the correct choice was made, the computed root values were quite good.

THE INPUT POLYNOMIAL COEFFICIENTS. THEY ARE IN ORDER IF READ IN ROWS.

0.1000000000000000D 01 -0.5000009999999999D 01 0.9000040000000000D 01 -0.7000050000000000D 01 0.2000020000000000D 01

```

*****
*
*      *** THE COMPUTED ROOTS OF THE POLYNOMIAL ***
*
*****
*
*      POLYNOMIAL REMAINDER WHEN
*      ROOT WAS DIVIDED OUT
*
*      REAL PART      IMAGINARY PART
*
*****
*
*      1      *      0.100000666680D 01      0.      *      -0.333066907388D-15      0.      *
*      2      *      0.100000666680D 01      0.      *      0.188737914186D-14      0.      *
*      3      *      0.200000600000D 01      0.      *      0.      0.      *
*      4      *      0.999996666400D 00      0.      *      0.      0.      *
*
*****

```

Exact polynomial:

$$P(x) = (x - 1)^2(x - 1.00001)(x - 2)$$

Comment: This problem is similar to the previous one. In this case the wrong choice between $x = 1.000000$ and $x = 1.0000067$ was made for the location of the multiple root. Of course, the root values are not as good as they would have been if the correct decision could have been made; but note that the centroid of the three roots in the group is quite accurate.

```
0.100000000000000000 01 -0.400000000000000000 01 0.6000001999999999 01 -0.400000400000000000 01 0.100000200000100000 01
```

$$P(x) = (x - 1 + 0.001i)^2(x - 1 - 0.001i)^2$$

75

THE INPUT POLYNOMIAL COEFFICIENTS. THEY ARE IN ORDER IF READ IN ROWS.

0.100000000000000000 01 -0.800000000000000000 01 0.250000019999999999 02 -0.400000120000000000 02 0.35000024000001000 02
-0.16000020000003999 02 0.30000060000002999 01

```

*****
*                                     *
*   *** THE COMPUTED ROOTS OF THE POLYNOMIAL ***   *
*                                     *
*****
*          *          *          *          *          *          *          *          *          *
* RCCT *          *          *          *          *          *          *          *          *          *
* NUMBER *          *          *          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *          *          *
*****
*          *          *          *          *          *          *          *          *          *
* 1 *          *          *          *          *          *          *          *          *          *
* 2 *          *          *          *          *          *          *          *          *          *
* 3 *          *          *          *          *          *          *          *          *          *
* 4 *          *          *          *          *          *          *          *          *          *
* 5 *          *          *          *          *          *          *          *          *          *
* 6 *          *          *          *          *          *          *          *          *          *
*          *          *          *          *          *          *          *          *          *
*****

```

RCCT NUMBER	REAL PART OF ROOT	IMAGINARY PART OF ROOT	POLYNOMIAL REMAINDER WHEN ROOT WAS DIVIDED OUT
	REAL PART	IMAGINARY PART	
1	0.9990765817760 00	0.	0.2442490654180-14
2	0.9996877264080 00	0.1487630401670-02	-0.3314258028320-13
3	0.9996877264080 00	-0.1487630401670-02	-0.3430589146090-13
4	0.3000000000000 01	0.	-0.1998401444330-14
5	0.1000773982700 01	0.9538712732370-03	0.
6	0.1000773982700 01	-0.9538712732370-03	0.

Exact polynomial:

$$P(x) = (x - 1)(x - 1 + 0.001i)^2(x - 1 - 0.001i)^2(x - 3)$$

Comment: This is the previous problem with a real root between the complex roots. If the program could get the multiple root first, the resolution would be quite good; but the program is structured to find real roots first. Since the polynomial was found to change sign in the vicinity of the root group near $x = 1.0$, a real root is known to exist. The real root could not be resolved with the ratio of derivatives theory so a root was taken out near the center of the group.

```

0.100000000000000000 01 -0.          -0.1000001000001000D 07 -0.          0.1000001000001000D 07
-0.          -0.100000000000000000D 01

```

Exact polynomial:

$$P(x) = (x + 0.001)(x - 0.001)(x + 1)(x - 1)(x + 1000)(x - 1000)$$

Comment: This problem shows good root accuracy over a wide range of root size. This is accomplished by (1) polynomial scaling and (2) extracting the lower magnitude roots first.

APPENDIX D

OUTLINE OF THE BASIC LOGICAL STEP PROCESS FOR THE RATIO OF DERIVATES ANALYSIS FOR REAL ROOTS

- (1) First x try at a given root
ND = 2
If $|P(1)| < \text{TOLRQ}$ and $|P(2)| < 10 \times \text{TOLR}$, ND = ND + 1.
- (2) Each new x try at a given root
 - (a) $\text{ARATIO} \leq 0.45$ (Appear to be at right root multiple, (ND - 1), and zeroing in.)
 - (1) If $|\text{RATIO}(\text{ND} - 1)| > 5/\text{TOLR}$, too close to the root for good analysis.
Back away to get in the $1/\text{TOLR}$ to $5/\text{TOLR}$ range.
 - (2) If $1/\text{TOLR} < |\text{RATIO}(\text{ND} - 1)| < 5/\text{TOLR}$, make the final root correction and divide the root out of the polynomial.
 - (3) If $1/\text{TOLR} > |\text{RATIO}(\text{ND} - 1)|$
 - (a) $\text{LIMIT} = 0$, ($|P(\text{ND} - 1)|$ is valid) make x correction by $1/\text{RATIO}(\text{ND} - 1)$ to get closer to the root.
 - (b) $\text{LIMIT} = 1$, ($|P(\text{ND} - 1)|$ is not valid. It is set to TOLRQ , and it is assumed to have greater magnitude than the actual value.) Set $\text{KTRIG} = 1$.
 - (1) If $|\text{RATIO}(\text{ND} - 1)| < 1000$, use present x value as root
 - (2) If $|\text{RATIO}(\text{ND} - 1)| > 1000$, step away from the present x to a point where a valid $\text{RATIO}(\text{ND} - 1)$ is reached so the $1/\text{RATIO}(\text{ND} - 1)$ final correction can be made to give a better root value.
 - (b) $0.45 < \text{ARATIO} < 1.0$. (Either at too low ND or too far from the root yet.)
 - (1) If $\text{ND} < \text{NDEG}$. Increase ND by 1 and come through the analysis again at the same x .
 - (2) If $\text{ND} = \text{NDEG}$. Not yet near enough to root. Step closer by $1/\text{RATIO}(\text{ND})$ correction. Set $\text{JTRIG} = 2$.
 - (c) $\text{ARATIO} \geq 1.0$. (In general, between roots.) From figures 2 and 3, it can be seen that root candidates occur on either side of such an x location. Set $\text{IMX} = 1$ to indicate the first of the two is being investigated by stepping along the real axis with LTRIG the step counter. Step with doubled step size until:
 - (1) $\text{ARATIO} < 0.45$
 - (a) $\text{LIMIT} = 0$. This point is a root. Do normal analysis.
 - (b) $\text{LIMIT} = 1$. Check the other point of symmetry in the effort to try to determine which is the root.

(2) LTRIG = 8

(a) IMSURE = 0. Give up the search for a real root here.

(b) IMSURE = 1. A real root is known to exist in this region. Continue stepping until a XHIGH or XLOW limit is reached and then step in the other direction.

REFERENCES

1. Hildebrand, Francis B.: Introduction to Numerical Analysis. McGraw-Hill Book Co., Inc., 1956.
2. Ralston, Anthony: A First Course in Numerical Analysis. McGraw-Hill Book Co., Inc., 1965.
3. McCracken, Daniel D.; and Dorn, William S.: Numerical Methods and FORTRAN Programming. John Wiley & Sons, Inc., 1964.